



**SOLUCIÓN INSTALACIONES AUTOMATIZADAS SOBRE SISTEMAS  
OPERATIVOS WINDOWS**

**UNIVERSIDAD CATÓLICA DE COLOMBIA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN  
BOGOTÁ D.C  
2017**

**SOLUCIÓN INSTALACIONES AUTOMATIZADAS SOBRE SISTEMAS  
OPERATIVOS WINDOWS**

**DARWIN ARNULFO LEGUIZAMÓN QUINCHE      CÓDIGO: 625091**

**ALEXANDER ARTURO VENGOECHEA GUERRERO      CÓDIGO: 625084**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE  
INGENIERO DE SISTEMAS**

**DIRECTOR**

**DIEGO ALBERTO RINCÓN YÁÑEZ MCSc**

**INGENIERO DE SISTEMAS**

**UNIVERSIDAD CATÓLICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

**BOGOTÁ D.C**

**2017**



## Atribución 2.5 Colombia (cc BY 2.5 co)

Este es un resumen legible por humanos (y no un sustituto) de la [licencia](#).

[Advertencia](#)



### Usted es libre para:



Compartir — copiar y redistribuir el material en cualquier medio o formato

Adaptar — remezclar, transformar y crear a partir del material

Para cualquier propósito, incluso comercialmente

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia

### Bajo los siguientes términos:



**Atribución** — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**No hay restricciones adicionales** — Usted no puede aplicar términos legales ni medidas tecnológicas que restrinjan legalmente a otros hacer cualquier uso permitido por la licencia.

### Aviso:

Usted no tiene que cumplir con la licencia para los materiales en el dominio público o cuando su uso esté permitido por una excepción o limitación aplicable.

No se entregan garantías. La licencia podría no entregarle todos los permisos que necesita para el uso que tenga previsto. Por ejemplo, otros derechos como relativos a publicidad, privacidad, o derechos morales pueden limitar la forma en que utilice el material.

## **NOTA DE ACEPTACIÓN**

Aprobado por el comité de grado en cumplimiento de los requisitos exigidos por la Facultad de Ingeniería y la Universidad Católica de Colombia para optar al título de Ingenieros de Sistemas.

---

Jurado

---

Diego Alberto Rincón  
Director

---

Revisor Metodológico.

17 de noviembre de 2017

## **AGRADECIMIENTOS**

Agradecemos principalmente a la Universidad Católica De Colombia por permitirnos culminar esta gran etapa en nuestras vidas profesionales.

A nuestras familias que con su esfuerzo y dedicación lograron educarnos con valores para llegar a ser los grandes profesionales que ellos han buscado que seamos.

A nuestro director de proyecto de grado, ya que sin su acompañamiento y guía no habríamos logrado culminar este último paso.

## INDICE DE CONTENIDO

INTRODUCCIÓN .....	12
1. GENERALIDADES .....	13
1.1. ANTECEDENTES .....	13
1.2. PLANTEAMIENTO DEL PROBLEMA .....	15
1.3. DESCRIPCIÓN DEL PROBLEMA.....	16
1.4. DELIMITACIÓN.....	17
1.4.1. ALCANCE.....	17
1.4.2. LIMITACIONES. ....	17
2. OBJETIVOS DEL PROYECTO.....	18
2.1. OBJETIVO GENERAL .....	18
2.2. OBJETIVOS ESPECÍFICOS .....	18
3. MARCO DE REFERENCIA .....	19
3.1. ESTADO DEL ARTE .....	19
3.1.1. Software Libre.....	19
3.1.2. Consideraciones para el despliegue de software. ....	20
3.1.3. Instalaciones de Software.....	21
3.1.4. Despliegue seguro de Software.....	22
3.2. MARCO CONCEPTUAL .....	24
3.2.1. NFS .....	24
3.2.2. Protocolo eXternal Data Representation .....	24
3.2.3. Protocolo SMB.....	24
3.2.4. Automatizador de Scripts (Auto IT).....	25
3.2.5. Servidor Web.....	26
3.2.6. Servidor de archivos .....	27
3.2.7. Servidor de base de datos.....	27
3.2.8. Directorio Activo.....	28
3.2.9. Arquitectura Cliente Servidor .....	28
4. METODOLOGÍA .....	31
4.1. METODOLOGÍA XP.....	31
5. DESARROLLO DEL PROYECTO .....	33
5.1. PLANIFICACIÓN DEL PROYECTO.....	33
5.1.1. Análisis de requerimientos. ....	33

5.1.1.1.	Requerimientos funcionales. ....	34
5.1.1.2.	Requerimientos no funcionales .....	35
5.1.1.3.	Casos de uso. ....	36
5.1.2.	Diseño de la solución. ....	37
5.1.2.1.	Diagrama de arquitectura. ....	37
5.1.2.2.	Diagrama de componentes. ....	39
5.1.2.3.	Diagrama de despliegue. ....	40
5.2.	IMPLEMENTACIÓN. ....	41
5.2.1.	Entorno .....	41
5.2.2.	Desarrollo. ....	42
5.2.3.	Fase de pruebas. ....	47
6.	RESULTADOS OBTENIDOS .....	52
7.	CONCLUSIONES .....	54
8.	TRABAJOS FUTUROS. ....	55
9.	BIBLIOGRAFIA. ....	56

## INDICE DE IMÁGENES

Ilustración 1 - Diagrama de Casos de uso .....	36
Ilustración 2 - Diagrama de arquitectura .....	37
Ilustración 3 - Diagrama de componentes .....	39
Ilustración 4 - Diagrama de despliegue .....	40
Ilustración 5 - Visual Studio MVC .....	43
Ilustración 6 - Visual Studio Recursos .....	43
Ilustración 7 - Grupos de administración del directorio activo .....	44
Ilustración 8 - Capa de Modelo.....	45
Ilustración 9 – Clase Usuario.cs .....	45
Ilustración 10 - Capa de Vista .....	46
Ilustración 11 - Dashboard de usuario usuario.cshtml.....	46
Ilustración 12 - Capa de controlador .....	47
Ilustración 13 - Validación de rol de usuario DashboardController.cs .....	47
Ilustración 14 - Login no válido.....	47
Ilustración 15 - AD Permisos PRY_APP_ADMINIS.....	48
Ilustración 16 – Login Rol Administrador .....	48
Ilustración 17 - Perfil administrador .....	49
Ilustración 18 - Opción Rutas .....	49
Ilustración 19 - Opción Crear paquete.....	50
Ilustración 20 - AD Permisos Usuario.....	50
Ilustración 21 - Login Rol Usuario .....	51
Ilustración 22 - Perfil Usuarios.....	51
Ilustración 23 - Función instalar software .....	52
Ilustración 24 - Descargando Aplicación .....	52
Ilustración 25 - Ejecutar aplicación automatizada .....	52
Ilustración 26 - Despliegue de aplicación .....	53
Ilustración 27 - Notificación de instalación completa .....	53
Ilustración 28 - Reportes de instalaciones realizadas .....	53



## **INDICE DE ANEXOS**

ANEXO A - Encuesta.....	61
ANEXO B – Documento de especificación de requerimientos. ....	62
ANEXO C – Documento de diseño.....	77
ANEXO D – Documento de arquitectura.....	94

## **ABSTRACT**

The degree project was carried out to solve a problem of repetitive processes in medium and large companies. This proposal, raises a system of control and administration of software in companies, from an unattended way. Initially, information was collected about the needs that exist in relation to the repetitive processes that were carried from the IT area, the process that was selected to automate the software installations was in Windows environment; that is because in most medium-sized and large companies, this kind of processes are carried out manually regarding several reasons. Based on the needs found, a more formal process was developed with the help of the requirements specification document to define the functional and non-functional requirements of the solution. Then, an architecture was designed that would define the best structure to carry out the project, by using the three-layer model, then the application design process is carried out and agile programming XP is defined as the most adequate way to proceed with coding stage. Finally, a software control and administration system is produced allowing users to install new programs of an unattended way, which allows to carry out a control of the applications that are installed by machine or by user from the web platform.

**Keywords:** Windows, Automatization, Installations, Software Deployment, Development, Services, Scripts, Role.

## RESUMEN

El proyecto de grado fue realizado para dar solución a un problema de procesos repetitivos en empresas de mediano y gran tamaño. Esta propuesta, plantea un sistema de control y administración de software en empresas, desde una intervención desatendida. Inicialmente se realizó el levantamiento de información acerca de las necesidades que existen frente a los procesos repetitivos que se realizan desde el área de IT, el proceso que se seleccionó para automatizar fue el de instalaciones de software en sistemas operativos Windows ya que en la mayoría de empresas de mediano y gran tamaño se realiza como un procedimiento manual a raíz de diferentes motivos. A partir de las necesidades encontradas, se realizó con ayuda del documento de especificación de requerimientos, un proceso más formal para definir los requerimientos funcionales y no funcionales de la solución. Luego, se diseñó una arquitectura que definiría la mejor estructura para llevar a cabo el proyecto, al utilizar el modelo a tres capas, se realiza el proceso de diseño de la aplicación y se define la programación ágil XP como la más adecuada para proceder con la etapa de codificación. Finalmente se obtiene un sistema de control y administración de software que permite a los usuarios instalar nuevos programas de forma desatendida, asimismo llevar un control de las aplicaciones que se instalan por maquina o por usuario desde la plataforma web.

**Palabras Claves:** Windows, Automatización, Instalaciones, Despliegue de Software, Desarrollo, Servicios, Scripts, Rol.

## INTRODUCCIÓN

El despliegue o instalación de aplicaciones en ambientes Windows, es un proceso que se contempla bajo la responsabilidad y atención de un usuario. Esta tarea requiere un accionar paso a paso (en el mejor de los casos) y un seguimiento constante. Sin embargo; ¿Qué ocurre si el usuario de una organización necesita instalar más de una aplicación?, ¿Qué ocurre si el usuario necesita hacer la misma cantidad de instalaciones o más en diferentes equipos de cómputo de la empresa?, pues ciertamente en el entorno empresarial es muy común realizar tareas repetitivas como instalar, desinstalar, actualizar y distribuir software. En efecto, este tipo de actividades en un ambiente no automatizado requiere del uso de recursos como personal y tiempo dedicado solo para procesos de este tipo, contando también con el tiempo que lleva coordinar la disponibilidad del cliente que solicita dicha aplicación. Asimismo, la necesidad de realizar instalaciones en diferentes equipos de cómputo se vuelve reiterativa debido a la rotación de personal, cambios en la forma de trabajar de los usuarios finales o daños que requieren reemplazar la máquina. Generalmente, las organizaciones disponen de personal calificado para realizar dichas acciones y para ello se requiere de conocimientos básicos en cuánto a instalación de software y comprensión del entorno empresarial en el que se encuentra.

El despliegue de aplicaciones no siempre es tan sencillo como seguir las instrucciones que guían a un usuario en el proceso de instalación, sino que se deben conocer necesidades reales de fondo, por lo que se requiere el control de las instalaciones de programas. Por ejemplo, un usuario final es capaz de descargar un software de internet e instalarlo de acuerdo con su necesidad, pero sin precisar la fuente; esto puede ser adverso para una compañía en caso de que el usuario final carezca de conocimientos respecto a seguridad. De esta manera el usuario pasaría por alto las autorizaciones y advertencias para las descargas de software, esto finalmente podría comprometer la integridad o la eficiencia del sistema con el despliegue de un malware, además de llevar a sanciones por instalación programas sin la debida autorización con su licencia como se indica en el texto de McGuire et al., (2002).

Teniendo en cuenta que se plantea la automatización de instalaciones como alternativa, en efecto la optimización se reflejaría en el tiempo dedicado por parte del personal de tecnologías de la información (IT), consecuentemente podrían resolver situaciones que agreguen valor a su servicio ante la compañía; por lo tanto se podría cuestionar ¿Cuánto le cuesta a una compañía tener personal dedicado a una labor específica?; sin duda alguna, todo el costo que abarcan las responsabilidades puntuales y que en contraste puedan ser vistas como un retorno de inversión, siempre darán gran valor para una compañía.

## 1. GENERALIDADES

### 1.1. ANTECEDENTES

El tema de la automatización de instalaciones no es nuevo. Con el pasar del tiempo y con el cambio constante en las tecnologías de la información, se ha hecho necesario tomar acciones que vayan encaminadas a solucionar patrones repetitivos, que faciliten el desarrollo de otros procesos que aportan valor a una organización. En este ámbito existen muchas soluciones propuestas que solventan sólo una parte de toda la temática expuesta anteriormente, por ejemplo, soluciones que sólo apuntan a registrar la trazabilidad de instalaciones en los equipos en una red o soluciones que tienen como objetivo desplegar solamente instaladores de programas.

En el mercado existe software tanto licenciado (de pago), como gratuito que permite realizar el despliegue de aplicaciones, sin embargo, estas no siempre se acoplan adecuadamente a las necesidades de control o de optimización de recursos. Por lo general, el software licenciado incluye la mayoría de funcionalidades esperadas, mientras que las soluciones gratuitas pueden ser aprovechadas para construir una solución más robusta y ajustada a una necesidad puntual. A continuación, se mencionan algunos productos que se encuentran en el mercado con funcionalidades para automatización y control de despliegue de software.

- **Microsoft System Center Configuration Manager (SCCM):** La información encontrada en (Corporation, n.d.), muestra que esta solución está creada por Microsoft para dar solución tanto al control como a la administración de hardware y software. Está enfocada al sector empresarial ofreciendo servicios de operations manager (administrador de operaciones), service manager (administrador de servicio), orchestrator (orquestador) y data protection manager (administrador de protección de datos). Hoy en día está en su versión 2016 la cual trae servicios para control de dispositivos móviles e integración con directorio activo de Azure.
- **Manage Engine Despliegue de Software:** La solución de automatización para despliegue de software creada por Manage Engine, se basa según su sitio web (manageengine, 2017), en la distribución de software a una alta cantidad de usuarios. Comprende una implementación que basada en un repositorio central de instaladores que serán reutilizados al ser lanzados desde un administrador central hacia varios clientes. Esta plataforma es parametrizable a diferentes tipos de usuario, ejecutando por cada despliegue la función (run As). Ofrece un visualizador de estado de proceso para validar si la instalación fue exitosa.
- **Total Software Deployment:** El desarrollo de la empresa SoftiventeLab, (2017), es una solución que provee servicios de despliegue masivo de software, con esta utilidad promueven a la automatización de procesos manuales, haciendo que desde un administrador central se realicen los despliegues de software y escaneo a través de la red para capturar la

información de los clientes. Esta herramienta cuenta con un robusto módulo de inventario enfocado a los activos de la empresa para control de cambios y depreciación de los elementos de tecnología.

Los criterios para elegir una solución se dan de acuerdo con reglas de negocio operativas, que tienden a dar valor en términos de disminución de costos. Las necesidades principales de una organización en este aspecto, se basan en controlar el estado de las instalaciones en su entorno, para fines de cumplimiento legal, agilidad auto-gestionando la instalación de paquetes de software, con aprobación previa por parte de la organización o la del departamento de tecnologías de información, controlar el despliegue por cantidad de paquetes instalados en los equipos, de acuerdo al licenciamiento de los diferentes programas y organizar un inventario de software utilizado por los diferentes departamentos o secciones de la compañía.

El software licenciado tiene restricciones y limitaciones además de ser de código propietario; muy seguramente si se necesitan personalizaciones se incurrirá en costos adicionales. Por otro lado, el software gratuito puede integrar funcionalidades para que se entregue una solución más ajustada a la necesidad de las reglas de negocio operativas; es por eso por lo que se deben tener en cuenta la mayor cantidad de soluciones granulares, que aporten a una construcción de una solución más grande, que cumpla con los requerimientos del cliente.

La pregunta que muchas organizaciones se formulan frente a este tipo de situaciones es: ¿Cómo abordar una solución que tenga el mejor costo – beneficio?

De la pregunta anterior se plantea que, un buen sistema se puede construir con varias herramientas libres, que permitan una ganancia rápida respecto a las necesidades organizacionales. Con soluciones de código abierto se permite ser escalable en este tipo de proyectos e inclusive se puede llegar a cubrir más necesidades de las previstas, también permite participar activamente en el perfeccionamiento de la solución, al estar disponible el código fuente a más personas calificadas.

## **1.2. PLANTEAMIENTO DEL PROBLEMA**

Hoy en día, existen herramientas que permiten preparar un despliegue inicial de máquina preestablecida con sistema operativo Windows. Pero: ¿Cómo automatizar instalaciones posteriores a una preparación inicial?

Las herramientas de preparación inicial nos permiten empaquetar y desplegar instalaciones justo después de la configuración del sistema operativo, mientras que otras nos permiten realizar la automatización de paquetes de forma individual.

El departamento de tecnologías de la información tiene la necesidad ante una compañía de optimizar tiempos de respuesta para mejorar la prestación del servicio a usuarios finales. Otra necesidad del departamento de IT es la de minimizar los riesgos en seguridad; si se logra permitir que los usuarios accedan a un repositorio central de software autorizado, evitaría las descargas desde fuentes desconocidas, ya que el acceso a esta clase de sitios podría comprometer los ordenadores con ataques cibernéticos. Por consiguiente, se entregaría autonomía a los usuarios para instalar software, sin la necesidad de disponer del tiempo del personal del departamento IT. Con lo anterior se lograría mejorar la percepción del servicio, además de dar espacio al personal técnico para estar dedicado proactivamente, a resolver problemas que puedan ser prioritarios para mantener disponible los servicios que mantienen la operación del negocio.

Entonces, ¿Cómo integrar dicho servicio con “las reglas de negocio” básicas de una empresa para controlar las instalaciones?

### 1.3. DESCRIPCIÓN DEL PROBLEMA

Se ha contemplado una muestra estadística (Véase Anexo A) en donde demuestra que, en un gran porcentaje las empresas siguen utilizando tiempo y mano de obra para realizar instalaciones de software. Ahora, desde el punto de vista interno del área de tecnología empresarial, se logra encontrar en las muestras tomadas, que la indisponibilidad laboral para el usuario final durante la instalación de software es una realidad.

Normalmente, el tiempo contemplado para que un usuario obtenga el software requerido bajo condiciones normales, depende en un 50% de la disponibilidad del área de sistemas y el otro 50% de la disponibilidad del usuario final. De manera que, el área de IT se ve altamente impactado por las métricas en sus acuerdos de nivel de servicio (SLA), sobre todo cuando las variables no son totalmente dependientes de su gestión y la percepción en los aspectos de servicio disminuye.

Se propone un caso común, partiendo de la necesidad de un usuario final ante alguna tarea o servicio necesario para cumplir con sus funciones. Dicha situación desencadena una serie de sucesos progresivos; en una situación normal el usuario debe solicitar dicha necesidad por medio de plataformas adaptadas para la solicitud de licencias, software, soporte, etc. El generar el ticket o el requerimiento llevaría unos minutos hacerlo, dependiendo de los campos requeridos por estas plataformas. El ciclo del proceso apenas va en ejecución y aún el departamento encargado de gestionar esta clase de requerimientos no está enterado. Cuando por fin el requerimiento es asignado al área encargada, el usuario debe esperar hasta 24 horas a que su solicitud sea autorizada, adicionalmente debe contar con la disponibilidad de tiempo de algún agente de IT que se encargue de tomar ese requerimiento, buscar el instalador, coordinar el momento y el modo de realizar dicha instalación, que bien puede ser en sitio o remoto.

Aunque el usuario final evidentemente es el más afectado cuando realiza requerimientos de software, es claro que la percepción no es tan mala como se pretendería y es debido a que este flujo de trabajo es común en las compañías, pero sí de alguna manera, el tiempo se pudiera optimizar para que el usuario logre tener su software, en el mismo tiempo que se demora en realizar la solicitud, con esto lograría entregar un ahorro importante al reducir los protocolos del proceso.



## **1.4. DELIMITACIÓN**

Con la delimitación se busca dimensionar el proyecto, considerando los siguientes aspectos:

### **1.4.1. ALCANCE.**

Los siguientes criterios serán determinantes para definir el proyecto:

#### **1.4.1.1. Tiempo.**

- El periodo de tiempo de implementación es menor a 6 meses.

#### **1.4.1.2. Contenido.**

- El prototipo permitirá desplegar, instalar y configurar software de forma desatendida.
- El despliegue de la solución permitirá llevar un control de las instalaciones realizadas y guardará la información en un sistema de base de datos.

#### **1.4.1.3. Espacio.**

- El proyecto se realizará en los hogares de cada uno de los integrantes del grupo, además se utilizarán los espacios que brinda la universidad para el proceso de seguimiento.

### **1.4.2. LIMITACIONES.**

La consideración de las limitaciones se basa en los siguientes aspectos:

#### **1.4.2.1. Espacio.**

- Todo el ciclo de diseño, desarrollo e implementación se realizará desde los hogares de cada uno de los integrantes del grupo, lo cual implica que la conexión a internet con re-uso, esto puede contar con limitaciones por velocidad moderada.

#### **1.4.2.2. Tiempo.**

- La implementación del prototipo se llevará a cabo en el segundo semestre del año 2017.
- El tiempo de desarrollo está limitado por el periodo académico.
- La actualización o retroalimentación de la solución se ve comprometida por tiempo definido.

## **2. OBJETIVOS DEL PROYECTO**

### **2.1. OBJETIVO GENERAL**

Diseñar una arquitectura de servicio orientada al despliegue automatizado de instalaciones en ambientes Windows.

### **2.2. OBJETIVOS ESPECÍFICOS**

- Analizar las principales necesidades empresariales para llevar a cabo el despliegue automatizado de software.
- Diseñar la arquitectura del sistema de información de acuerdo con el análisis realizado.
- Desarrollar el prototipo que implemente las funcionalidades principales para solucionar las necesidades encontradas en el análisis realizado.
- Ejecutar una fase de pruebas que permita evaluar la calidad de la solución.

### **3. MARCO DE REFERENCIA**

#### **3.1. ESTADO DEL ARTE**

Hoy en día, en el modelo empresarial según González, Santos, Vargas, Martín-Gutiérrez, & Orihuela, (2013) se propone la tecnología como una herramienta para optimizar procesos que normalmente serían empleados por mucho recurso humano o que son realizados de manera repetitiva.

Partiendo de esta realidad se hace necesario conocer en detalle las variables que podrían estar involucradas tanto en el desarrollo de software, tipos de software, licenciamiento, transporte de aplicaciones, etc.

En el entorno del desarrollo, se considera al software libre como una alternativa por el ahorro de costos a una compañía, sin embargo, es necesario definir los términos y consideraciones que se deben tener al momento de decidir qué tipo de licenciamiento utilizar. A continuación, se entrará en el detalle de la terminología y la descripción de cada uno de estos factores.

##### **3.1.1. Software Libre**

Para 1984 era imposible usar un programa para ordenadores sin primero instalar un sistema operativo propietario, el cual se encargaría de administrar los recursos, este debía ser obtenido según el autor del libro (Rosa & Heinz, 2007) bajo una licencia de uso restrictiva. Ningún tipo de usuario tenía permiso para compartir libremente el software con otros, además el código fuente estaba restringido para cambios o prácticamente para cualquier tipo de personalización para sus propias necesidades.

El software libre se define por su tipo de licenciamiento por lo que se puede llamar “Software licenciado bajo condiciones libres”, así fue nombrado según el autor del libro (Chinchilla-Arley, 2011) para efectos de terminología.

El software libre nace en 1983 como un proyecto llamado el GNU3 (Sistema operativo UNIX) publicado por Richard Stallman, (2004), cuando era investigador del Laboratorio de Inteligencia Artificial del Massachusetts Institute of Technology. Stallman inició este proyecto con la intención de crear un sistema operativo compatible con Unix, pero siempre con el enfoque de código abierto, de forma tal que cualquier persona pudiera hacer uso, bien fuera para modificarlo o adaptarlo a sus propias necesidades. Su orientación consistía en “promocionar la libertad de los usuarios de ordenadores, y defender los derechos de los usuarios de software libre” (Stallman, 2004).

La Fundación de Software Libre (Free Software Foundation) publica la primera versión de GNU como licenciamiento libre, a la cual llamaron General Public License (GPL) durante el año de 1989 según la información de Cataldi & Salgueiro, (2007), es allí en donde nace el concepto de licenciamiento libre, el cual está orientado a “garantizar la libertad de compartir y modificar todas las versiones

de un programa, con el fin de asegurarse de que sigue siendo software libre para todos sus usuarios” textualmente indicado así por Stallman, (2004) .

Es importante aclarar que el software gratuito no es sinónimo de software libre, existe gran variedad de software gratuito, el cual es posible descargar y utilizar sin ninguna restricción; sin embargo, no se tiene acceso al código fuente, por lo tanto, no puede ser modificado o adaptado.

El software libre se define como aquella aplicación que es adquirida sin intercambio lucrativo, a su vez es suministrada junto con el código fuente, el cual puede ser copiado, distribuido, modificado y adaptado a las necesidades de los distintos tipos de usuarios que lo requieran indica Chinchilla-Arley, (2011).

El despliegue empresarial, es un poco más específico según los autores Capeáns Hurtado & Rodríguez Puente, (2015) y enfocan las llamadas libertades para el uso y que se deben cumplir para que sea software libre.

Las libertades llamadas así por González et al., (2013) indica en principio que, el uso del programa con cualquier propósito debe ser respetado, en efecto se puede adaptar a sus propias necesidades y por lo tanto debe contar con el acceso al código fuente; así mismo puede distribuir copias a otros interesados para recibir ayuda complementaria. Finalmente se tiene un panorama muy amplio en la descarga o utilización del software y su adaptación a nuestras necesidades de automatización.

En pocas palabras, el software libre es fundamental cuando de ahorro de costos se trata, ahora bien, si se enfoca el tipo de licenciamiento a un ambiente libre, se logra por naturaleza encontrar soporte y ayuda con la comunidad, finalmente creando una red completa de ayuda a nivel mundial en el tema de interés. A continuación, se indicará el enfoque basado en las consideraciones que requiere el despliegue de software en una perspectiva general, destacando el uso del software libre.

### **3.1.2. Consideraciones para el despliegue de software.**

Desplegar software para cualquier tipo de uso, requiere de una estrategia para lograr la adopción de los nuevos métodos. La estrategia debe estar basada en la gestión del cambio y enfocada principalmente en las personas que necesitan de este según Macías Sánchez, (n.d.). Este enfoque facilitará el proceso de transición a los cambios que implica el despliegue de las nuevas implementaciones y permitirá minimizar la resistencia a dichos cambios.

Se ha detectado que la mayoría de las investigaciones en mejora de procesos, están enfocadas en la parte tecnológica, pero pocas mencionan otros factores importantes como la cultura, la gestión del cambio, las personas, la comunicación, la formación durante y después del proceso de despliegue. No obstante, los factores humanos para la mejora de procesos software han sido ignorados y esto ha impactado fuertemente la percepción del desarrollo conceptual de automatizar el proceso mecánico. Ahora bien, la falta de adecuación de propuestas conforme a las necesidades de cualquier ambiente, es un factor determinante frente al fracaso

de las iniciativas, a razón de que las soluciones no suplen de las necesidades, según indica Carlos, Rojas, Ariel, & Alegría, (2007) el autor como critica al desarrollo de nuevos modelos para automatización.

En otros aspectos se debe tener en cuenta como el despliegue de software se realiza bajo un proceso de instalación, a su vez es necesario conocer los tipos de instalaciones que existen actualmente y como se diferencian.

### **3.1.3. Instalaciones de Software**

#### **3.1.3.1. ¿Qué es una instalación?**

Una instalación está definida en estos artículos (Charles J. Gazdik, Kelli Kinkela, 1998; Kim Douglas Mason, 2006) como, el proceso por el cual un nuevo sistema operativo, un programa, o un driver, son copiados al disco duro, después el mismo se despliega y ejecuta un proceso en el cual los componentes son utilizados, para que según su arquitectura sean ubicados en donde se requieren según sus características de uso. Los fabricantes de software buscan simplificar al máximo el proceso de instalación, que implica la configuración y cualquier otra actividad necesaria para ese fin según lo mencionado por Capeáns Hurtado & Rodríguez Puente, (2015).

La instalación de un programa se entrega en un paquete comprimido que contiene un set de archivos necesarios, ya que con esto se puede manejar mejor la distribución y venta.

Antes de instalar el archivo deberá ser descomprimido en la ubicación determinada por el usuario o por el sistema operativo, usualmente estas ubicaciones son carpetas de archivos temporales, ya que el uso de estos archivos solo será necesario mientras se realiza la instalación y no es óptimo almacenarlos en el equipo si solo se usaran durante este proceso.

#### **3.1.3.2. Tipos de instalaciones**

Las definiciones que se mencionan a continuación se basan en la referencia de Kralik, (n.d.), en donde se definen claramente las diferencias entre los diferentes tipos:

- **Instalación Limpia:** Este tipo de instalación se presenta cuando se realiza por primera vez en el sistema, es decir, nunca se ha realizado dicha instalación y así se puede prevenir fallos por instalaciones del mismo programa en ocasiones previas.
- **Instalación Atendida:** Es el tipo de instalaciones más comunes en los sistemas operativos Windows. Este tipo de instalaciones requieren la presencia del usuario quien manualmente tendrá que seguir con el proceso de instalación. Los asistentes ofrecen al usuario la elección de aceptación o negación del contrato de usuario final (EULA), además de elegir la ubicación

de la instalación, configuración de contraseñas, todo esto a través de una interfaz gráfica.

- **Instalación Silenciosa:** Es el método más usado para el despliegue del Malware, usualmente se confunde el término con la instalación desatendida pero no son lo mismo, toda instalación desatendida es silenciosa, más no toda instalación silenciosa es desatendida. En definición este tipo de instalaciones se realizan sin que el usuario se entere que se está realizando ya que no muestra mensajes de ejecución, proceso o finalización, solo después de cierto tiempo el usuario se da cuenta que lo tiene, bien sea por que requiere de algún complemento o porque el rendimiento de la maquina se ve afectado al ser un malware el que se desplegó.
- **Instalación Desatendida:** El método de instalación desatendida tiene lugar cuando se realiza sin supervisión de un usuario final o cuando no requiere la atención del usuario. Una instalación al ser desatendida excluye la parametrización personalizada ya que los parámetros estarán definidos antes de ser desplegado el instalador.
- **Instalación de Bajo Nivel:** Este tipo de instalaciones son todas aquellas que se llevan a cabo sin una interfaz gráfica por lo cual se ejecuta de forma tal que optimiza el tiempo de ejecución al momento del despliegue.
- **Instalación Programada:** Las instalaciones programadas son aquellas que se realizan a través de un plan de ejecución por tiempo, es decir, se desplegaran sin iniciativa del usuario, se ejecutarán después de un tiempo determinado por el usuario o el sistema operativo al no poderse realizar el proceso en ese momento. También se conoce como instalación planeada.
- **Instalación en Red:** Otra forma de realizar instalaciones es a través de la red en donde con contenedores de información en este caso instaladores, se puede realizar un despliegue de la instalación copiando los datos a una ubicación temporal para poder realizar el proceso sin problemas y no usando el almacenamiento local de la maquina destino. Este proceso es muy útil cuando se requiere un mismo software en muchas maquinas que estén interconectadas a través de una misma red LAN.

En resumen, las instalaciones de software a nivel técnico poseen protocolos y normas que ya se han mencionado, pero al mismo tiempo se debe considerar la seguridad de su despliegue, acto seguido se procede a especificar qué características existen para que el despliegue de software sea seguro.

#### **3.1.4. Despliegue seguro de Software**

En el entorno empresarial moderno, se tiene relativamente una gran cantidad de computadores asignados, usan una típica estructura administrativa que permite al administrador supervisar varios equipos desde una ubicación central. Dentro de las actividades que le corresponden al administrador podrían estar tareas como, desplegar aplicaciones a cada uno de los equipos, instalar parches o versiones correctivas para mantenimiento o simplemente actualizaciones que podrían ser de aplicaciones ya instaladas; asimismo la desinstalación de software obsoleto para optimizar el rendimiento de las maquinas, en casos comunes de soporte en las mesas de ayuda a los usuarios finales. El administrador del sistema podría

optimizar el tiempo y los procesos usando procesos automatizados de bajo nivel desde un servidor administrador teniendo conectados varios host conectados a la red según muestra en el desarrollo del trabajo publicado por J.-A. Wang & Liu, (2003).

Todo instalador convencional de software tiene un ayudante que vuelve intuitivo el proceso para el usuario y finalmente termina por instalar el software. Este ayudante o Wizard consiste en una serie de pasos que solicitan la interacción por parte del usuario en cada uno de esos pasos. Dependiendo del software este brinda al usuario diferentes opciones a través de cajas de diálogos e instalaciones que no pueden proseguir, hasta que no sean ejecutadas las instalaciones de software que son prerequisite para que este funcione óptimamente lo cual fue detallado por Manzoor & Nefti, (2010).

Las técnicas para la seguridad en el despliegue de software son descritas en una implementación, se puede ver como un método que incluye al preparar el paquete instalador del software, visto como un dispositivo de ambiente red. A lo que se refiere (paquete de instalación) un almacenamiento completo incluyendo las políticas asociadas con el paquete del software, estas son preparadas y publicadas en el paquete de instalación (Y. Wang & Rowe, n.d.). Por otro lado, se generó otra opción, en cuanto a las políticas de instalación dando la opción de crear un archivo alterno para aplicar este tipo de políticas sobre el instalador, esto incluye permisos y características específicas (J.-A. Wang & Liu, 2003).

## **3.2. MARCO CONCEPTUAL**

### **3.2.1. NFS**

Las siglas NFS (Network File System o Sistema de Archivos de Red) desarrollado por SUN Microsystems en el año 1984, referencia tomada de (Profesorado, 2006). Su función en una red permite que un equipo GNU/Linux pueda montar y trabajar con un sistema de archivos de otro equipo de la red como si fuera local.

El servicio NFS utiliza las llamadas a procedimientos remotos basadas en el protocolo RPC (Remote Procedure Call), que permite desde un equipo (cliente) ejecutar código ubicado en otro equipo remoto (servidor), mediante el establecimiento de sockets (IP + puerto) entre ambas indicado por Instituto Nacional de Tecnologías Educativas y Formación del Profesorado, (2017).

#### **3.2.1.1. Cómo funciona NFS**

Con respecto a la necesidad de subir un directorio exportado desde un servidor NFS, se utilizaría la orden mount, indicando el tipo de sistema de archivos, la maquina remota (en donde está instalado y configurado el servidor NFS) y el directorio a cargar en el punto de montaje local.

La orden Mount conecta con el protocolo RPC.mountd, que está ejecutándose en la maquina remota vía RPC. Es entonces, cuando el servidor comprueba si la máquina que hace la petición del montaje tiene permisos sobre el directorio raíz. En caso de tenerlos se lleva a cabo la ejecución como si se tratara de un dispositivo físico en la maquina (hard disk, cd/dvd, USB, etc.) según lo indicado por el autor de (Profesorado, 2006).

### **3.2.2. Protocolo eXternal Data Representation**

El protocolo eXternal Data Representation (XDR) es la especificación para una representación estándar de varios tipos de datos definido así por Center, (2006).

Mediante la utilización de una representación de tipo de datos estándar, un programa puede estar seguro de interpretar los datos correctamente, incluso si el origen de los datos es una máquina con una arquitectura completamente diferente.

En relación con la práctica, la mayoría de programas no utilizan XDR internamente; en su lugar, utilizan la representación de tipo de datos específica de la arquitectura del sistema con motivo de ejecutar el programa, visto que el programa necesita comunicarse con otros, convierte los datos al formato XDR antes de enviarlos y de igual manera lo hace a la inversa cuando recibe datos, simultáneamente los convierte de formato XDR a su propia escritura de tipo de datos específico (Center, 2006).

### **3.2.3. Protocolo SMB**

El protocolo del bloque de mensajes del servidor (SMB) es, un protocolo de uso compartido de archivos de red en donde permite que las aplicaciones de un equipo, puedan leer y escribir archivos adicionalmente solicitar servicios desde los



programas de un servidor en una red de equipos detallado en la definición de Corporation, (2012). El protocolo SMB, si bien por un lado puede usarse sobre el protocolo TCP/IP, también puede una aplicación (o el usuario de una aplicación), acceder a los archivos u otros recursos, permitiendo que las aplicaciones puedan leer, crear y actualizar archivos en un servidor remoto. Así pues puede comunicarse con cualquier programa del servidor que esté configurado para recibir la solicitud de un cliente SMB, acorde con la información encontrada en los sitios de (Corporation, 2012)(IBM, n.d.).

### **3.2.4. Automatizador de Scripts (Auto IT)**

AutoIT es un programa diseñado para automatizar procesos que surgen en la configuración de equipos para usuarios finales. Esta herramienta simula tanto las pulsaciones del teclado como los movimientos del cursor del mouse permitiendo que el modelo básico de secuencia de acciones sea realizado de un modo totalmente desatendido para el usuario final, esta información se detalla en (AutoIT, n.d.).

A pesar de que AutoIT comenzó funcionando como una herramienta de automatización simple, ahora tiene funciones y características que le permiten ser utilizado como un lenguaje de scripting de propósito general.

Las características del lenguaje incluyen:

- Elementos habituales de alto nivel para funciones, bucles y análisis de expresiones.
- Una gran cantidad de funciones de manipulación de cadenas y un motor de expresión regular.
- Soporte COM.
- Llamados a Win32 y API de DLL de terceros.

Este programa ha sido diseñado para ser tan ligero y autónomo como sea posible, sin archivos .dll externos o entradas de registro necesarias para que sea seguro para usar en Servidores. Los scripts pueden ser compilados en ejecutables autónomos con Aut2Exe como lo explica Flesner, (2007) y AutoIT, (n.d.).

También proporciona una versión COM y DLL combinada de AutoIT llamada AutoITX que le permite agregar las características únicas de AutoIT a sus propios scripts favoritos o lenguajes de programación, como lo indican Brand & Balvanz, (2005).

Los beneficios que ofrece la herramienta según la información encontrada en (SCITE, n.d.), son:

- Editor incorporado con validador de sintaxis para script.
- Independencia en su ejecución y sistema liviano.
- Compatibilidad con Unicode y versiones de x64 en los componentes
- Simulación en combinación de teclas y movimientos del ratón.
- Gestión de ventanas
- Gestión de controles

- Interfaces graficas

### 3.2.5. Servidor Web

Para fines del proyecto hablaremos de dos puntos de vista enfocados al servidor web, el primer enfoque será hacia la definición y funcionalidades de un servidor web; el segundo enfoque será la descripción y características de Internet Information Server el cual será clave para el desarrollo del proyecto.

**3.2.5.1. Definición de un servidor web:** Un Servidor Web se encarga básicamente de enviar y procesar solicitudes y peticiones que, son enviadas desde un cliente hasta un sistema que se encarga de procesar dicha solicitud, a su vez al realizar dicho proceso el servidor se encarga de transmitir la respuesta y comunicársela al cliente. Este tipo de solicitudes se realizan a través de un navegador o Browser, por el lado del servidor se usa el protocolo HTTP (Hyper Text Transfer Protocol) y se comunican a través de una conexión TCP, lo cual indica que este tipo de protocolo será de solicitud y respuesta tipo cliente / servidor como lo explica Duplika, (n.d.)

La función principal de un servidor web se basa en almacenar archivos del sitio y emitirlos a través de la red, para poder estar disponible para los usuarios o clientes. Hay que tener en cuenta que los servidores web solamente prestan servicios básicos de almacenamiento y transmisión de datos, mas no cuenta con propiedades de seguridad ni otros aspectos que se encuentran en un servidor de aplicaciones según lo resalta en el texto Mejías, Cintado, (n.d.).

**3.2.5.2. Servidor Web IIS (Internet Information Server):** IIS es la versión de servidor web desarrollada por Microsoft Corporation, el cual proporciona una plataforma de administración muy amigable con un estilo modular y extensible en donde se pueden alojar sitios web, servicios y aplicaciones de manera confiable.

Entre las características entregadas por la empresa desarrolladora Microsoft, (2016) se encuentran las siguientes ventajas de su implementación que serán relevantes para el prototipo propuesto:

- La seguridad web se refuerza gracias a una superficie reducida de servidor y al aislamiento automático de aplicaciones.
- Se logra el aislamiento de aplicaciones al proporcionar a los procesos de trabajo una identidad única y una configuración en espacio aislado de manera predeterminada, lo que reduce aún más los riesgos de seguridad.
- Podrá agregar y eliminar componentes IIS integrados e incluso reemplazarlos fácilmente por módulos personalizados que se adapten a las necesidades del cliente.

- Aumenta la velocidad del sitio web mediante el almacenamiento en caché dinámico integrado y la compresión mejorada.

### 3.2.6. Servidor de archivos

Un servidor de archivos se utiliza como un repositorio centralizado de archivos que son distribuidos dentro de una red de computadores. El servidor cumple con la función principal de permitir que se efectúe el acceso remoto a los archivos almacenados o que son directamente accesibles por otros nodos, esta información fue tomada de (Microsoft corp, n.d.)

Las características que tiene un servidor de archivos son:

- 3.2.6.1. Servicio:** Según el autor Bolosky et al., (1996) el sistema NFS comprende dos partes principales, la primera es un servidor y la segunda serán los clientes que requieren el acceso al servicio. Los clientes accederán de forma remota a los archivos que se encuentran directamente en el servidor.
- 3.2.6.2. Centralización y control:** Microsoft corp,( n.d.) indica que los usuarios no requieren tener disponible un directorio raíz en cada uno de los computadores de la organización, en otro caso solo necesitaran crear dicho directorio en el servidor NFS para posteriormente acceder a ellos a través de la infraestructura de red.
- 3.2.6.3. Recursos compartidos:** Los equipos clientes optimizarán el espacio en disco, dado que los archivos se encontrarán centralizados en un único lugar con la ventaja de concurrencia, de tal forma que no es necesario replicar información lo que nos garantiza que no halla redundancia en los datos, esta información la indican Yikang Xu, Uresh K. Vahalia, Xiaoye Jiang, Uday Gupta, (1999) y Bolosky et al., (1996) como complemento a las demás características.

### 3.2.7. Servidor de base de datos

Un servidor de base de datos es una aplicación que se fundamenta en la arquitectura cliente / servidor. Esta aplicación se compone de dos partes importantes, una llamada Front-end la cual se ejecuta desde el cliente y la otra llamada Back-end que se ejecuta desde el servidor designado para el almacenamiento de la información junto con el análisis de los datos (Tsirogiannis, Harizopoulos, & Shah, 2010).

MySQL es un gestor de bases de datos relacional, el cual permite almacenar una gran cantidad de información. Este gestor utiliza el lenguaje estructurado SQL, por lo tanto, implícitamente incluye un motor de almacenamiento InnoDB y ACID; entre sus funcionalidades están los procedimientos almacenados, triggers y vistas. Lo que facilita la administración de la información, así como los procesos que se quisieran implementar.

Una de las ventajas más relevantes de este gestor de base de datos es la denominada GPL (General Public Licence) como lo categoriza (Chinchilla-Arley,

2011), es decir que no tiene costo. Además de ser gratuito este gestor es soportado por Oracle hoy en día lo cual es una gran ventaja en su implementación.

Las características principales tomadas del autor Thibaud, (2006), de este administrador de base de datos son:

- Cuenta con el modelo ACID.
- Puede realizar varias operaciones simultáneas sobre la misma tabla.
- Permite administrar información importante como perfiles de usuarios y contraseñas, manteniendo altos niveles de seguridad.
- Implementa el uso del rol back, sub-consultas y transacciones.

### **3.2.8. Directorio Activo**

El directorio activo es una función que proporciona la capacidad de establecer información centralizada en una organización, este es un rol incluido en los servicios disponibles en las versiones de Windows server ("CAPITULO III," n.d.); con este servicio se simplifica ampliamente la administración de usuarios y equipos, proporcionando además la obtención de un acceso mejorado a recursos de red según especifica Desmond, (2008).

Como su nombre lo indica, es un servicio de directorio que entre sus características permite la adaptación a distintas bases de datos según sea requerido, lo cual es muy importante pues brinda la oportunidad de estandarizar el inicio de sesión, asimismo aplicar políticas globales o personalizadas para la administración de usuarios, permisos de acceso con el fin cumplir con los aspectos de confidencialidad de la información detallado por Dias, (2002). Otra funcionalidad relevante para el desarrollo del proyecto es la instalación centralizada de aplicaciones, este aspecto funciona como una política que permite a partir de un repositorio centralizado realizar instalaciones predeterminadas asociadas a perfiles o a usuarios, teniendo en cuenta para fines organizativos. Es importante mantener la estructura recomendada por Microsoft según Bonnet, (2014), para garantizar que el diseño de un esquema organizacional sea categorizado por grupos y con esto se logre la administración de las GPO (Group Policy Object); pensando en el control del cambio que se proporciona con frecuencia frente a la rotación de personal.

Las cuentas que gestiona el active directory se almacenan en una base de datos tipo SAM (Security Accounts Manager), lo cual beneficia su gestión puesto que no solo se puede almacenar la información de usuarios, sino que además permite guardar información relevante de estaciones de trabajo, servidores, recursos, directivas de seguridad entre otros como lo indica el manual de Allen & Hunter, (2006); y Chadwick, (2005).

### **3.2.9. Arquitectura Cliente Servidor**

Esta arquitectura consta de dos partes importantes la primera es el cliente y la segunda es el servidor. Estas partes básicamente actúan, siendo el cliente quien realiza peticiones al servidor sobre un programa, o algún servicio al que este le da respuesta. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes (Minera, 2008).

#### **3.2.9.1. Características de la arquitectura Cliente-Servidor**

Basados en las características explicadas por Guzman, Davila, Garcia, & Caicedo Rossi, (1998), podemos resaltar las que son mencionadas a continuación:

- El cliente facilita una interfaz para que el usuario interactúe con el sistema.
- El Servidor procede como proveedor del servicio en donde administra los recursos que el cliente utilizara para cumplir con una tarea determinada.
- Provee una facilidad para establecer relación entre los procesos, estos podrán ser ejecutados por diferentes maquinas a través de la red.
- La concurrencia es amplia dado que muchos clientes podrán realizar solicitudes al mismo tiempo sin bajar el rendimiento de la tarea para cada uno de los involucrados, aunque esto dependerá puntualmente de la red y de las características del servidor.
- Los clientes se denominan procesos activos ya que son los que realizan las peticiones de los servicios y los servidores se consideran como procesos pasivos ya que son los que reciben y ejecutan las peticiones.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ellos. El mensaje sirve como medio para realizar la petición y hacer entrega de los resultados frente a las solicitudes de servicios.
- El ambiente en el que se desempeñan se considera como heterogéneo, pues el cliente y el servidor no requieren tener el mismo sistema operativo para comunicarse. Una gran ventaja de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.

#### **3.2.9.2. Ventajas de la arquitectura Cliente-Servidor**

Teniendo claras las características ahora es momento de resaltar las ventajas que han sido especificadas por Casas González, (2008).

- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente-Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.
- Puede usar un raid de equipos económicos que cumplen con la misma función para evitar la adquisición de equipos más costosos que usualmente son requeridos para soluciones centralizadas.

- Puede usar equipos y software de distintos fabricantes lo cual apoya a la reducción de costos favoreciendo la flexibilidad en implementación de soluciones.
- Facilita la integración de nuevas tecnologías al usar estructura modular lo que hace que la infraestructura crezca favoreciendo la escalabilidad de las soluciones.
- Permite la integración de información proporcionando diferentes sistemas adecuados a cada departamento de la organización.

## **4. METODOLOGÍA**

El desarrollo de este prototipo se realiza usando la metodología XP, ya que es una de las más representativas de las metodologías ágiles existentes. Esta metodología se fundamenta en el trabajo en equipo como lo indica José H. Canós, (2012), donde su principal objetivo es mantener las buenas relaciones interpersonales pues esto conlleva a un cumplimiento óptimo, efectivo, eficaz de la labor encomendada y por supuesto de la meta asignada.

La metodología XP cuenta con retroalimentación continua entre el cliente y el equipo de desarrollo, manteniendo una comunicación clara y fluida entre los interesados del proyecto. XP se recomienda para proyectos que cuentan con requerimientos abstractos o que tienden a cambiar en poco tiempo mientras se modela la necesidad final, de esta forma se detalla por Bautista, 1999; Kasiak & Godoy, (2012).

### **4.1. METODOLOGÍA XP**

Esta forma de trabajo logra niveles de eficiencia y valor en el producto superiores a los obtenidos con otro tipo de ingeniería secuencial y de producción basada en procesos. Este método de desarrollo se usa para desarrollar nuevos productos realizados con equipos reducidos, multidisciplinarios, que trabajan con comunicación directa y empleando ingeniería concurrente, en lugar de ciclos o fases secuenciales.

Los objetivos que plantea esta metodología se acoplan para garantizar la calidad del software, haciendo que se puedan superar las expectativas del cliente. Por otra parte, establece el uso de buenas prácticas para los desarrollos de proyectos, en este caso la metodología canaliza las habilidades de los integrantes del equipo para mejorar la productividad.

Para emplear XP lo ideal es establecer un contexto claro sobre la dimensión del proyecto. Primero se debe descifrar bien al cliente, es decir se debe identificar a los interesados, también es importante contar con el equipo de trabajo idóneo el cual no podrá ser mayor a 12 integrantes y deben contar con características especiales tales como capacidad de aprendizaje continuo y un nivel de formación elevado. Finalmente dejar claridad en cuanto a los requisitos ya que pueden y serán cambiados conforme se van realizando las entregas.

Para el desarrollo del proyecto es necesario seguir algunas prácticas básicas de la programación extrema las cuales son descritas por Joskowicz, (2008) y José H. Canós, (2012), estas son explicadas a continuación:

- **Diseño Simple:** Se busca realizar lo preciso o ajustado de la manera más sencilla o liviana cosa tal que el código o las implementaciones se mantengan sencillas sin perder el enfoque de la funcionalidad.
- **Equipo Completo:** Se considera parte del equipo a toda persona interesada en el desarrollo del proyecto, esto incluye al cliente y al responsable del proyecto o director de proyecto.
- **Versiones Pequeñas:** Las versiones parciales que se entregan semanalmente son consideradas siempre y cuando le den valor al producto final. Las entregas deben ser tan pequeñas que se pueden entregar semanalmente.
- **Pareja de Programadores:** La metodología sugiere que se debe hacer grupos de a dos por cada ordenador de tal manera que tras un periodo de tiempo intercambien roles. Siempre debe estar uno codificando y el otro observando en pro de mantener un ritmo de trabajo sostenible y constante.
- **Ritmo Sostenible:** El ritmo deberá ser constante, lo que implica que no debe existir días muertos en los que no se avance con nada del proyecto, por el contrario, diariamente se deben plantear las metas con el fin de avanzar constantemente en la entrega parcial y final.
- **Integración continua:** Cada entrega parcial debe ser unificada, esto con el fin de evitar que la retroalimentación quede de lado mientras se realiza un trabajo que no sigue la continuidad. La idea es encontrar los errores más fácilmente lo que con el pasar del tiempo se convierte en algo más complejo.
- **Pruebas del Cliente:** El cliente acompañado por el equipo de desarrollo diseñaran un conjunto de pruebas que puedan probar las entregas parciales. Lo anterior se ajusta mucho a las pruebas de concepto.

Finalmente, con el método se consigue realizar una programación más organizada que permita tener un menor margen de error al momento de la entrega del producto final. También se tendrá la satisfacción del programador gracias al enfoque de la conducta que a metodología maneja como principio. Sin embargo, es de tener en cuenta que las técnicas se prestan para ser usadas en proyectos a corto plazo en consideración con la magnitud del mismo.



## **5. DESARROLLO DEL PROYECTO**

En el desarrollo del proyecto, se expondrá la transición del ciclo del sistema en donde se evidenciará como a partir del proceso de identificar los requerimientos se diseña una arquitectura, la cual se implementa con el fin de automatizar el despliegue de aplicaciones de forma totalmente desatendida, buscando finalmente llevar a la fase de pruebas en donde se validará tanto la calidad como el cumplimiento de las necesidades iniciales.

### **5.1. PLANIFICACIÓN DEL PROYECTO.**

#### **5.1.1. Análisis de requerimientos.**

El análisis que se realizó originalmente se basó en la situación actual de muchas empresas de mediano y gran tamaño. Inicialmente se hizo levantamiento de información por medio encuesta (Véase Anexo A) mediante la cual se pudo indagar con todo tipo de usuarios de diferentes empresas y cargos, las preguntas son muy puntuales frente a la percepción que tiene el usuario normal con el servicio entregado por parte del área de IT, enfocado hacia el modo en que se compone el proceso para solicitar y desplegar software. Las respuestas generan un concepto claro que demuestra la existente necesidad cuando de solicitar software se trata y más cuando las políticas de seguridad existen con la intención de mantener la integridad de los activos con el fin de evitar vulnerabilidades de seguridad informática en la operación del negocio; todas estas medidas se realizan con el fin de mejorar la apreciación del usuario en cuanto a la implementación de soluciones, que facilitan y automatizan procesos que hoy en día aún se siguen llevando a cabo de forma manual o con acompañamiento de usuarios autorizados del área de IT.

Como la intención es abarcar el proceso para el despliegue automatizado de software, se crea la necesidad de enfocar el proyecto a plataformas Windows únicamente, con esto se hace el levantamiento de las funcionalidades básicas que debe tener la solución y son las siguientes:

- El usuario final de la solución es aquel que, bajo su propia distribución del tiempo, puede solicitar instalar un programa de forma automatizada a través de la aplicación web presentada.
- Se asume que el usuario final no tiene conocimientos específicos en la forma de instalar un programa requerido para su labor.
- El usuario administrador de la solución es aquel que puede dedicar más tiempo atendiendo temas críticos y que no son “repetitivos” como las solicitudes de instalación de un programa.
- El usuario administrador puede supervisar a través de un registro histórico las solicitudes de instalación.

- Con una estructura adecuada de grupos de directorio activo, el usuario administrador puede administrar los permisos relacionados en la solución web sobre los diferentes paquetes de automatización.

Ya teniendo la idea de lo que se espera de la solución final, se procede a generar el listado de los requerimientos que serán el esquema del prototipo a realizar. Los requerimientos globales se enlistarán a continuación, sin embargo, el detalle se encontrará en el documento de especificación de requerimientos (Véase Anexo B).

#### 5.1.1.1. Requerimientos funcionales.

**Tabla 1 - Requerimientos funcionales**

Identificación	Nombre Requerimiento	Descripción
<b>RF01</b>	Ingreso al sistema	Para el proceso de ingreso se debe preguntar por usuario y contraseña para la correcta identificación de rol de usuario.
<b>RF02</b>	Consultar permisos asignados	La plataforma debe consultar permisos asignados a un usuario sobre los paquetes de automatización para exponer al mismo los paquetes permitidos.
<b>RF03</b>	Ejecutar paquetes de automatización	La plataforma deberá ser capaz de ejecutar un paquete de automatización en un servidor de archivos.
<b>RF04</b>	Crear registro de un paquete de automatización	La plataforma debe guardar registros que relacionen el instalador automatizado con grupos de directorio activo a los cuales pertenecen usuarios finales.
<b>RF05</b>	Leer grupos de directorio activo	La plataforma debe ser capaz de leer los grupos de directorio activo a los que pertenece un usuario final. Así mismo, desde el rol del usuario administrador, la plataforma debe ser capaz de devolver grupos de directorio activo sobre una OU determinada.
<b>RF06</b>	Generar reportes de instalaciones.	La plataforma debe generar reportes de instalaciones realizadas a través de la plataforma por usuario y por registro de paquete de instalación.
<b>RF07</b>	Verificar instalación en equipo local.	La plataforma debe verificar si la instalación de un programa solicitado por medio de un paquete

		de automatización ya se encuentra realizada en la máquina local.
--	--	--

### 5.1.1.2. Requerimientos no funcionales

Junto con los requerimientos funcionales, se establecen también los requerimientos no funcionales los cuales se generan para dar calidad en tiempo de ejecución al prototipo; estos están detallados en la siguiente tabla:

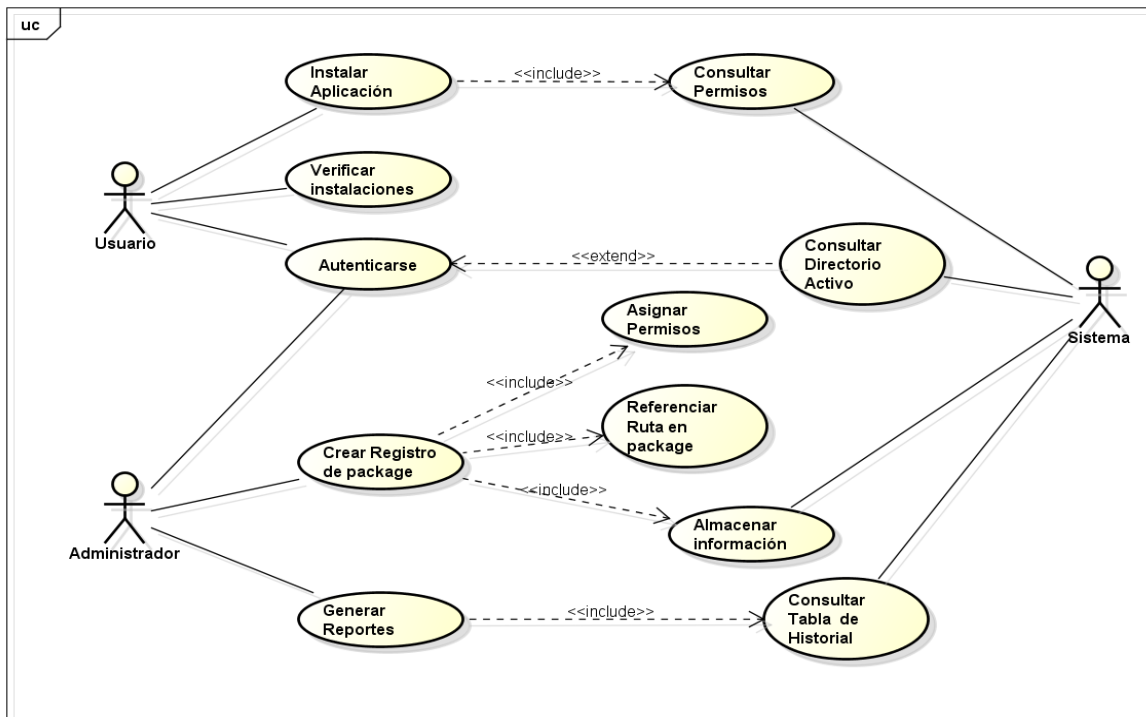
**Tabla 2 - Requerimientos no funcionales**

Identificación	Nombre Requerimiento	Descripción
<b>RNF01</b>	Interfaz gráfica	La interfaz gráfica de la plataforma debe ser sencilla y entendible de acuerdo con las funciones.
<b>RNF02</b>	Usabilidad	El aplicativo web de la plataforma debe ser sencillo, intuitivo y agradable para los usuarios
<b>RNF03</b>	Disponibilidad	La aplicación puede estar soportada por configuración de servicios sin alta disponibilidad.
<b>RNF04</b>	Desempeño	La plataforma deberá garantizar una respuesta rápida ante cualquier solicitud.
<b>RNF05</b>	Seguridad	La plataforma deberá garantizar la seguridad y confirmación de los datos de autenticación.
<b>RNF06</b>	Soporte	Se deberá describir la configuración de los servicios que intervienen en la plataforma.

Con los requerimientos ya establecidos, se procede a realizar el modelo de casos de uso que se implementa según el estándar UML (lenguaje unificado de modelado). En la siguiente imagen (Véase Ilustración 1 - Diagrama de Casos de uso) se pretende realizar un esquema grafico de los requerimientos anteriormente mencionados.

### 5.1.1.3. Casos de uso.

Ilustración 1 - Diagrama de Casos de uso



El diagrama de casos de uso refleja gráficamente los requerimientos funcionales del sistema, mostrando un flujo lógico de lo que debe ser considerado como la base de la instalación desatendida y automatizada de software. En el caso más relevante de la solución, se habla del proceso de despliegue de la aplicación, por lo cual el caso de uso que refleja dicho comportamiento será:

El Stakeholder (actor del sistema) llamado usuario instala una aplicación a la cual tiene permisos. Para llegar a esto debe pasar por la consulta de la información que se realiza desde otro de los Stakeholders llamado sistema. Este actor cumple con consultar los datos en el directorio activo cuando el usuario se autentica.

Los detalles adicionales a los mencionados anteriormente se podrán consultar en el documento de especificación de requerimientos (Véase anexo B).

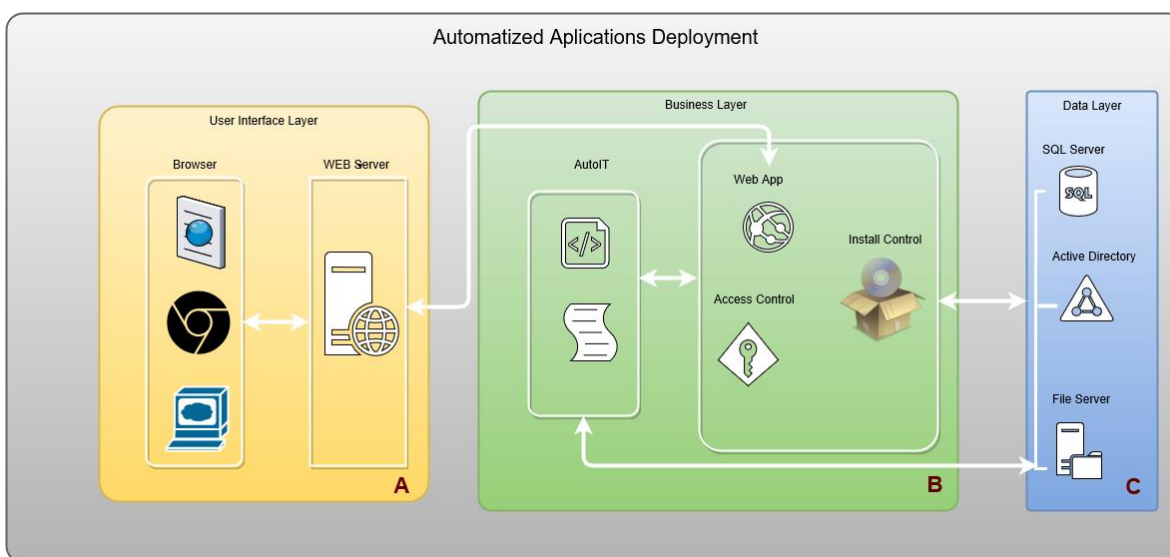
Ya teniendo claros los requerimientos, se definen los servicios y herramientas para llevar a cabo la construcción del prototipo. Para esto se debe establecer la arquitectura del sistema más conveniente, la infraestructura de los servicios necesarios para poder simular las suposiciones y dependencias que se describen en el documento de diseño (Véase anexo C), el lenguaje de programación a utilizar en la codificación, el diseño de datos, la relación que hay entre los datos y el sistema, entre otras variables que se generan en esta fase de análisis, pero serán desarrolladas en la fase de diseño.

### 5.1.2. Diseño de la solución.

Para el desarrollo de la fase de diseño, lo primero que se debe dejar claro es la arquitectura que se va a implementar. Tras dimensionar el alcance de la solución, se ha concluido que la arquitectura cliente servidor es la óptima para el desarrollo, a su vez, partiendo de los supuestos especificados en el documento de especificación de requerimientos (Véase Anexo B), se encontrara básicamente que para la implementación de la solución ya se debe contar con una infraestructura como la red LAN y con unos servicios implementados tales como un controlador de dominio y el directorio activo; esto es muy relevante para la determinación de enfocar el diseño de la solución en un modelo MVC (modelo, vista, controlador). Para explicar gráficamente el diseño final de la arquitectura, es necesario ver el diagrama de arquitectura (Véase **Error! Reference source not found.**).

#### 5.1.2.1. Diagrama de arquitectura.

**Ilustración 2 - Diagrama de arquitectura**



El diagrama se encuentra separado por las tres capas del MVC, en la parte A se hace referencia a la Vista, es decir todo lo que compete a la parte grafica de la aplicación con la cual el usuario tendrá interacción. En la parte de vista, se plantean dos tipos de elementos, el primero es un navegador que entrega la interfaz gráfica de forma amigable para los dos tipos de usuarios, los cuales se contemplaron en el documento de especificación de requerimientos (Véase Anexo B); por un lado, está el usuario normal el cual es el que solicita el software y por el otro está el administrador de la plataforma quien será el que otorgará los permisos para el despliegue de aplicaciones a los usuarios normales. La segunda parte, es el servidor web que está encargado de gestionar las peticiones que se realizan

desde el cliente y se ejecuta en la parte lógica de la aplicación, esta será detallada en la descripción del módulo de controlador.

Ahora, en el módulo B se encuentra toda la lógica del sistema, por un lado, está la persistencia de los datos, la cual se encarga de mantener en memoria toda la estructura de los permisos de cada usuario, este proceso es realizado desde la aplicación web. Además de realizar el proceso lógico la aplicación está encargada de mantener la integridad de los datos, como se contempló inicialmente los permisos para el uso de las aplicaciones se suministrarán a través del directorio activo, por lo tanto, la aplicación requiere realizar solicitudes continuamente al directorio activo por medio del protocolo LDAP.

Para la lógica de la aplicación, es necesario generar y almacenar scripts automatizados con las instrucciones secuenciales para cada programa que se desee instalar, es por tal razón que se implementa un módulo llamado AutoIT el cual será el encargado de esta función.

Si bien ya se habló de la vista y la lógica del modelo, ahora se debe contemplar el almacenamiento y la fuente de los datos. En el caso de esta solución los datos son paquetes de instaladores con su correspondiente script automatizado que serán almacenados en el File Server. Al realizar el proceso se debe contar con el registro de cada despliegue, por lo cual en el modelo de base de datos se almacenará el registro de seguimiento sobre las instalaciones, así poder generar los reportes y tener el control de las licencias para cada software. La única fuente de datos necesaria en la estructura de la aplicación es el directorio activo, allí se encontrará la distribución entre grupos de forma jerárquica y sobre ellos los permisos para cada tipo de usuario.

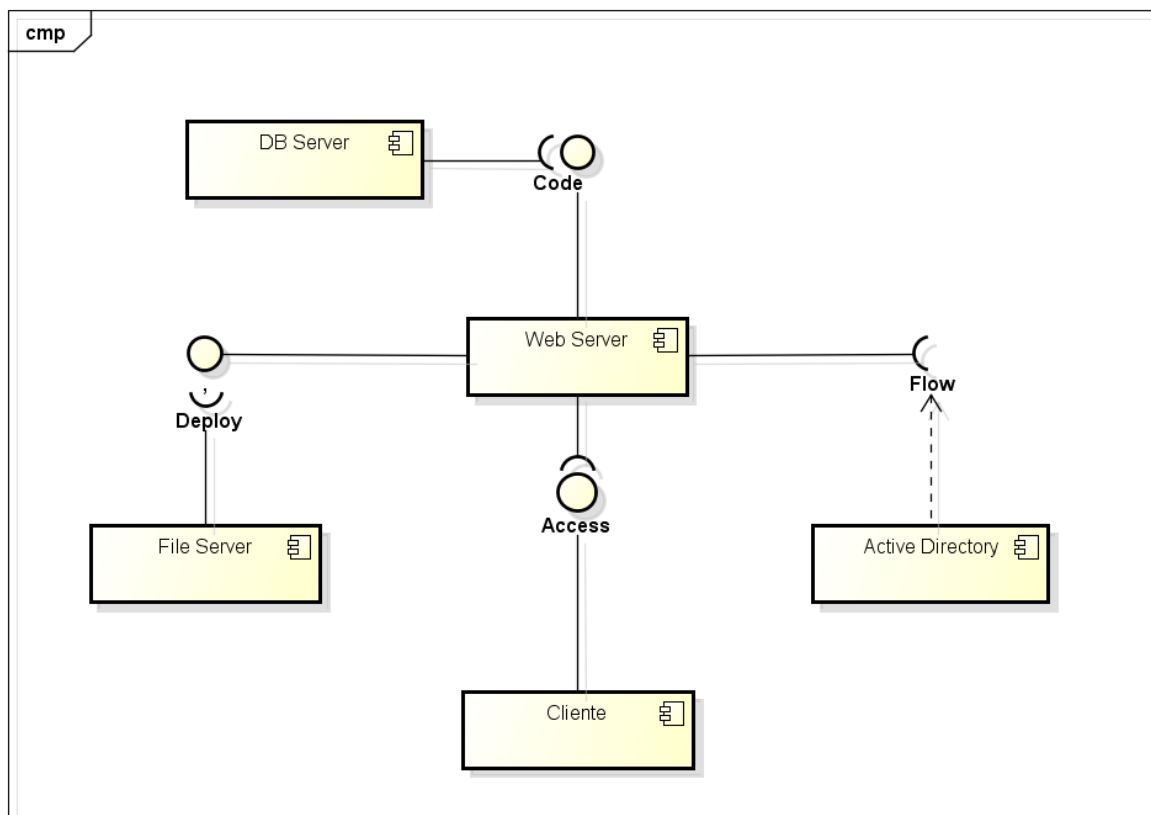
Partiendo de la arquitectura, los diagramas que serán más relevantes frente al diseño de la aplicación son:

- **Diagrama de componentes:** Este diagrama representa cada uno de los módulos que componen el sistema y como estos interactúan entre sí.
- **Diagrama de despliegue:** Este diagrama describe la interacción que hay entre los elementos básicos del sistema que pueden ser de hardware o software.

Adicionalmente a los diagramas mencionados, se han realizado diagramas de procesos BPMN, diagrama de datos entidad relación y de secuencia para mostrar en detalle cómo se plantea cada uno de los elementos que componen el sistema.

### 5.1.2.2. Diagrama de componentes.

Ilustración 3 - Diagrama de componentes

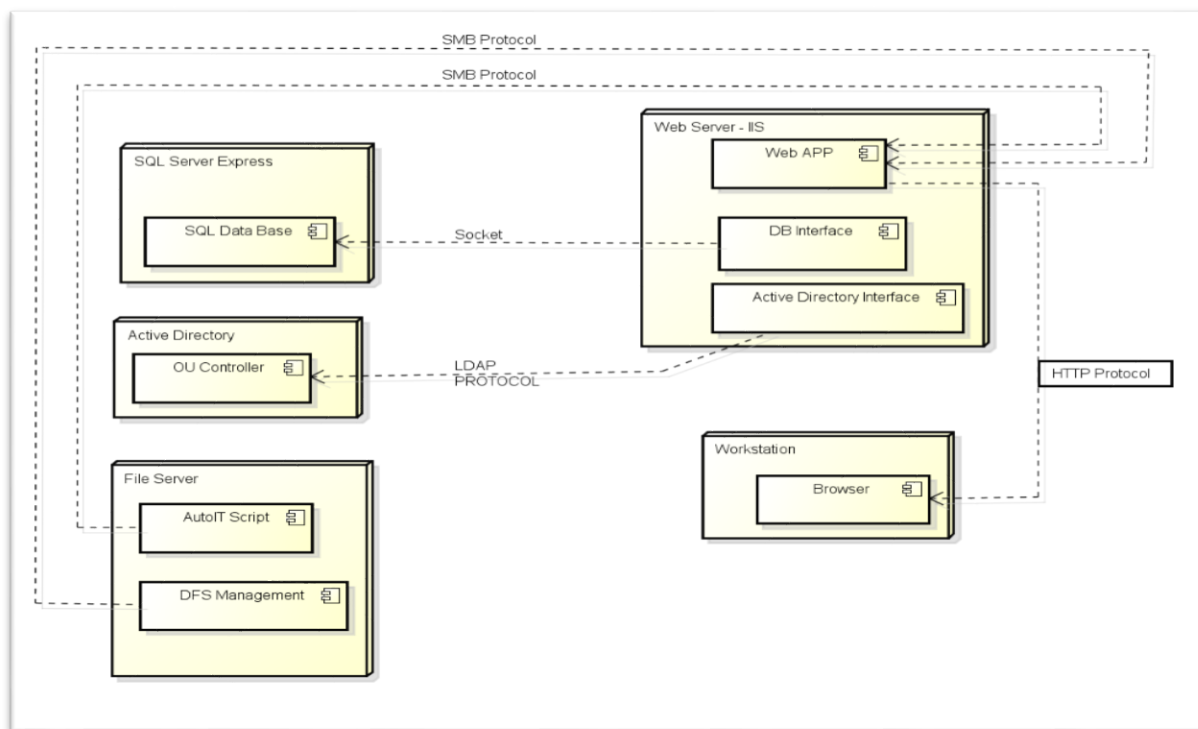


El diagrama de componentes muestra los distintos módulos que conforman el sistema propuesto, en el cual detallamos cómo interactúan los componentes en un entorno de ejecución.

Inicialmente parte desde el componente cliente quien es el que realiza la solicitud al web server usando la interfaz de acceso, esta interfaz se encarga de validar las credenciales con el módulo de active directory en un modo directo. Ya con las credenciales validadas, el componente web server se encarga de consultar en la base de datos los permisos que el cliente tiene sobre la solicitud enviada, al ser confirmado el proceso, este se trae desde el componente File server el script de automatización con su ejecutable y de esta manera la respuesta que entrega al cliente es el instalador parametrizado.

### 5.1.2.3. Diagrama de despliegue.

#### Ilustración 4 - Diagrama de despliegue



Con el diagrama de despliegue se pretende mostrar cada uno de los componentes de hardware y de software que componen el sistema, a su vez ilustra como esta distribuido cada modulo en los diferentes componentes de hardware que se requieren para poder llevar a cabo el despliegue de instrucciones de automatización.

El flujo comienza cuando, desde un computador (workstation) por medio de un web browser se accede a la plataforma y se realiza una solicitud, esta por medio del protocolo http llega a la aplicación que se encuentra alojada en el servidor web. Cuando el usuario accede a la plataforma, la aplicación web valida las credenciales del usuario accediendo a una instancia del directorio activo por medio del protocolo LDAP, cuando las credenciales ya estan validadas la aplicación consulta a la base de datos por medio de un socket de comunicación, los permisos que estan asociados al usuario. Finalmente cuando el usuario envia la solicitud de instalar por medio del mismo protocolo http, la aplicación se comunica con el file server por medio del protocolo SMB, en este se encuentran alojados el DFS (Distributed File System) y el AutoIT Script, entregando posteriormente la aplicacion automatizada y el package parametrizado para ejecutarse de modo desatendido como respuesta al usuario.



## 5.2. IMPLEMENTACIÓN.

### 5.2.1. Entorno

Para la implementación se realiza la definición de un entorno de servicios Windows, que es el ambiente objetivo para el público de destino del proyecto.

En la siguiente tabla, se encuentran descritos los servicios:

**Tabla 3 - Descripción de servicios del entorno**

<b>Nombre del servicio / elemento</b>	<b>Descripción / Propósito</b>
<b>Servicio de directorio activo</b>	Este servicio ha sido incluido en el entorno debido a que, a través del mismo, se realizarán las validaciones de autenticación de usuarios y de grupos que corresponden a los permisos de los usuarios sobre un paquete de automatización.
<b>Servicio motor de bases de datos</b>	En este, se referencia el modelo relacional de la lógica de la aplicación web. Las tablas que componen la base de datos y sus relaciones se encuentran detalladas en el documento de especificación de requerimientos (Véase Anexo B). El servicio aprovisionado es Microsoft SQL Server Express.
<b>Servidor de archivos</b>	Funciona como el repositorio del compilado de las automatizaciones y de los instaladores que puede leer y ejecutar la aplicación web. A través de un esquema de permisos, sólo se dará acceso a dichas rutas a un usuario de servicios único creado en el directorio activo con permisos de administrador sobre las máquinas clientes.
<b>Servicio DNS</b>	El servicio de nombres de dominio es utilizado para el registro de correspondencia entre nombres y direcciones IP. Con este servicio, se encuentran los diferentes servicios por nombre desde los diferentes componentes o servicios.
	Servicio de direccionamiento dinámico para los elementos pertenecientes al

<b>Servicio DHCP</b>	dominio. Sólo los servidores que tienen los servicios instalados tienen un direccionamiento fijo o asignado.
<b>Servicio de Controlador de dominio</b>	A este pertenecen tanto los servidores como los equipos clientes, simulando una red corporativa. Los elementos pertenecientes, por ejemplo, podrán leer grupos del directorio activo a través de la aplicación.
<b>Máquina cliente</b>	Corresponde a una máquina de usuario final, quién podrá acceder al aplicativo web desde dicha máquina e instalar una aplicación de acuerdo con los permisos determinados por un administrador de la plataforma.

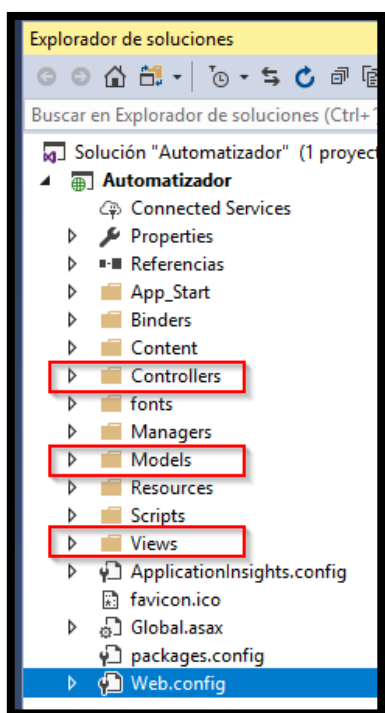
Los anteriores servicios o elementos son los básicos de un entorno empresarial con ambientes Windows. Con él, los elementos permiten ser identificados para la aplicación de políticas de seguridad y para accesos a los diferentes servicios del dominio empresarial.

### 5.2.2. Desarrollo

Para el desarrollo, se ha utilizado Microsoft Visual Studio Community 2017. La razón de ello es que dicha aplicación para desarrollo del entorno web, integra y reconoce implícitamente las conexiones con los servicios Windows.

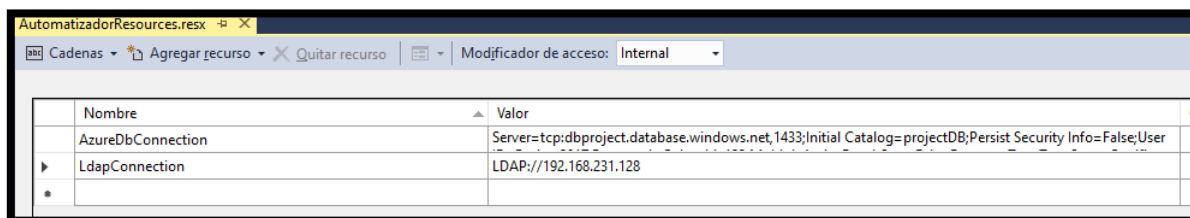
El modelo de arquitectura utilizado para la codificación es modelo, vista, controlador. Esto permite tener un control más adecuado definiendo las responsabilidades de cada capa.

## Ilustración 5 - Visual Studio MVC



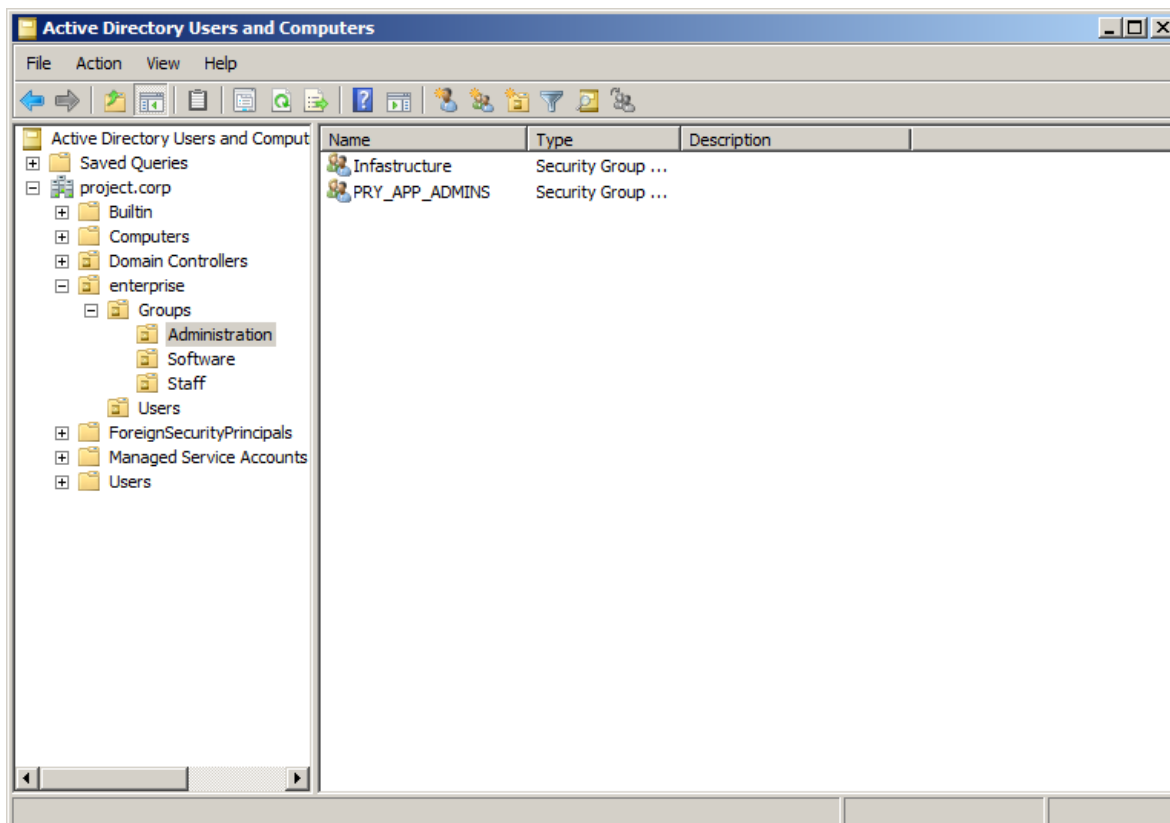
Para las integraciones entre sistemas, se han utilizado las interfaces de conexión al directorio activo y base de datos, que son primordiales para la lógica de la plataforma web. Para ello dentro de la solución del desarrollo, se ha organizado una categorización de recursos que permiten parametrizar las conexiones.

## Ilustración 6 - Visual Studio Recursos



Para la validación de vistas de cara a los usuarios de la aplicación web, se utiliza el grupo de directorio activo "PRY\_APP\_ADMINS" (Véase Ilustración 7 - Grupos de administración del directorio activo).

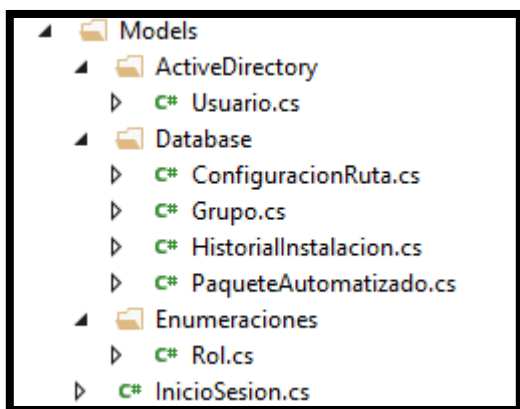
## Ilustración 7 - Grupos de administración del directorio activo



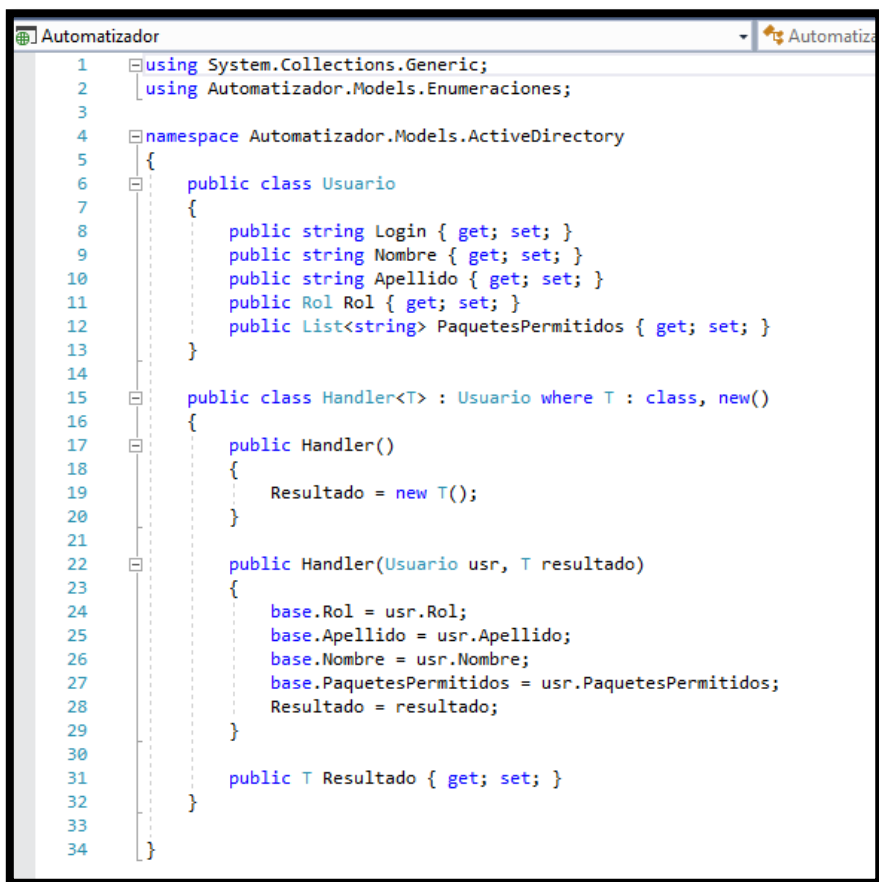
La pertenencia de un usuario en dicho grupo determinará si es administrador, si no pertenece, entonces es un usuario final (quien desea instalar una aplicación).

Dentro de la capa de modelo, se encuentran las abstracciones de los sistemas externos de base de datos y directorio activo que permitirán a través de clases definidas, extraer la información desde el sistema destino para ser validadas dentro de la aplicación.

## Ilustración 8 - Capa de Modelo

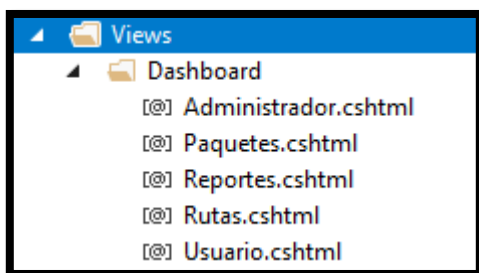


## Ilustración 9 – Clase Usuario.cs



En la capa de vista, se encuentran los diferentes formularios identificados para cada tipo de usuario según sus funciones dentro del sistema.

## Ilustración 10 - Capa de Vista

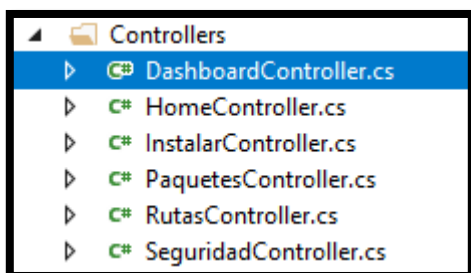


## Ilustración 11 - Dashboard de usuario usuario.cshtml

```
1  @using Automatizador.Models.Database
2  @model Automatizador.Models.ActiveDirectory.Usuario
3
4  @{
5      ViewBag.Title = "Dashboard Usuario";
6  }
7
8  <div class="jumbotron">
9      <h1>Bienvenido Usuario</h1>
10     <p class="lead">En esta seccion puede consultar los paquetes que le son permitidos instalar</p>
11 </div>
12
13 <table id="tablaDatos" class="table table-striped table-bordered" cellspacing="0" width="100%">
14     <thead>
15         <tr>
16             <th>Nombre</th>
17             <th>Versión</th>
18             <th>Accion</th>
19         </tr>
20     </thead>
21     <tbody>
22         @foreach (PaqueteAutomatizado paquete in ViewBag.Paquetes)
23         {
24             <tr>
25                 <td>@paquete.Nombre_Programa (@paquete.Nombre_Paquete) </td>
26                 <td>@paquete.Version</td>
27                 <td><a href="/instalar/programa/@paquete.Id_Paquete">Instalar</a></td>
28             </tr>
29         }
30     </tbody>
31 </table>
32
33 @section Scripts{
34     <script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
35     <script>
36         $(function () {
37             $('#tablaDatos').DataTable();
38         })
39     </script>
40 }
41
```

Cada uno de los formularios se integra con las capas modelo y controlador para extraer los datos de permisos correspondientes a cada usuario dependiendo de su rol.

## Ilustración 12 - Capa de controlador



## Ilustración 13 - Validación de rol de usuario DashboardController.cs

```
namespace Automatizador.Controllers
{
    [Authorize]
    public class DashboardController : Controller
    {
        private readonly DbManager _db = new DbManager();

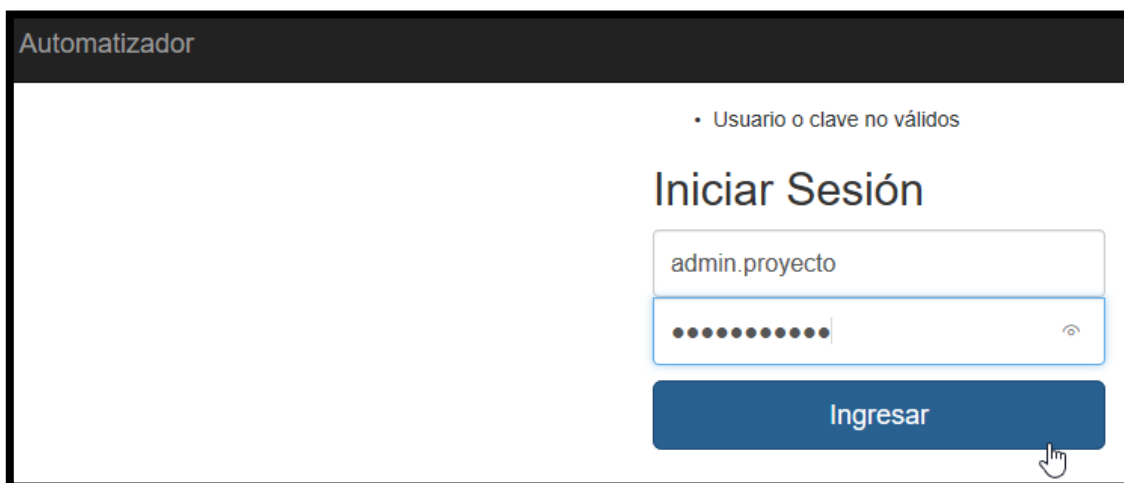
        public ActionResult Index(Usuario usr)
        {
            return RedirectToAction(usr.Rol == Rol.Administrador ? nameof(Administrador) : nameof(Usuario));
        }
    }
}
```

## Fase de pruebas.

Para la fase de pruebas, se comienza lanzando el formulario Login.cshtml en el navegador, a través del cual se validarán datos de rol del usuario que se autenticará.

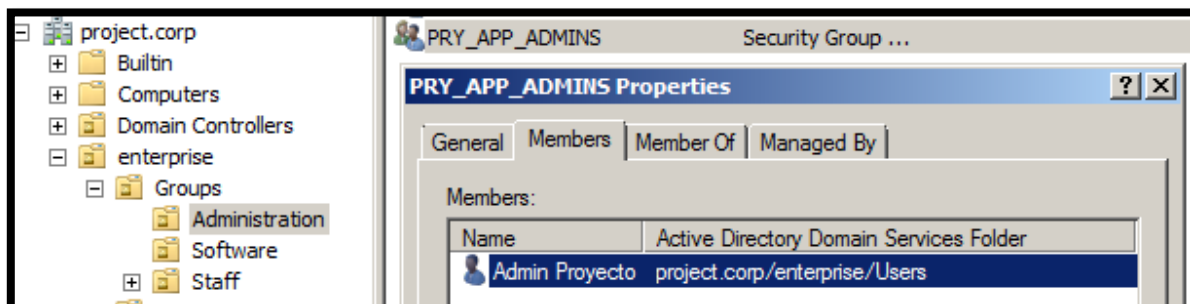
Para cualquier rol, la solución realiza validación en el formulario de autenticación de usuario y contraseña correctos contra objetos de usuarios existentes en el directorio activo:

## Ilustración 14 - Login no válido



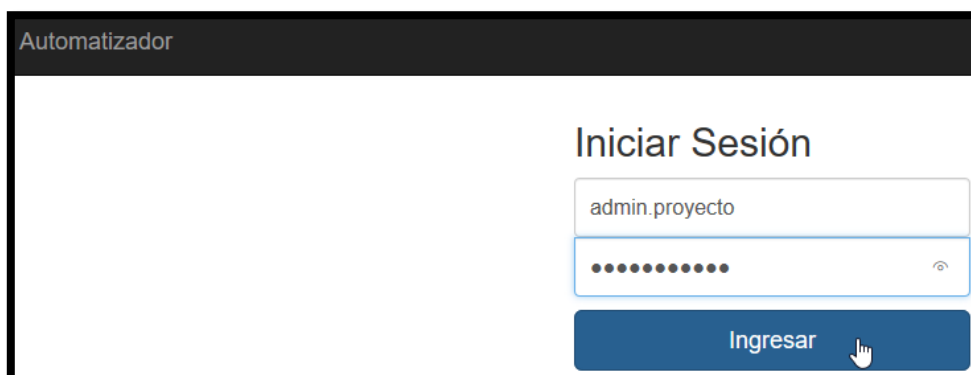
Dentro del grupo PRY\_APP\_ADMINS del directorio activo, se encuentra agregado el usuario admin.proyecto, que será el usuario de servicio administrador de la solución:

#### Ilustración 15 - AD Permisos PRY\_APP\_ADMINS



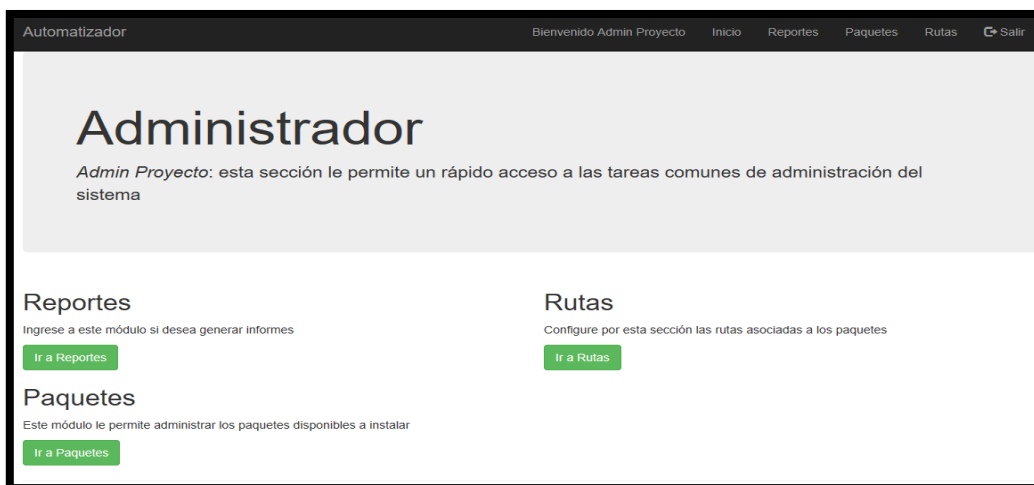
Lo anterior quiere decir que, si el usuario autenticado pertenece a dicho grupo, será redirigido al formulario de administrador, mientras que si no pertenece será redirigido al formulario de usuario.

#### Ilustración 16 – Login Rol Administrador



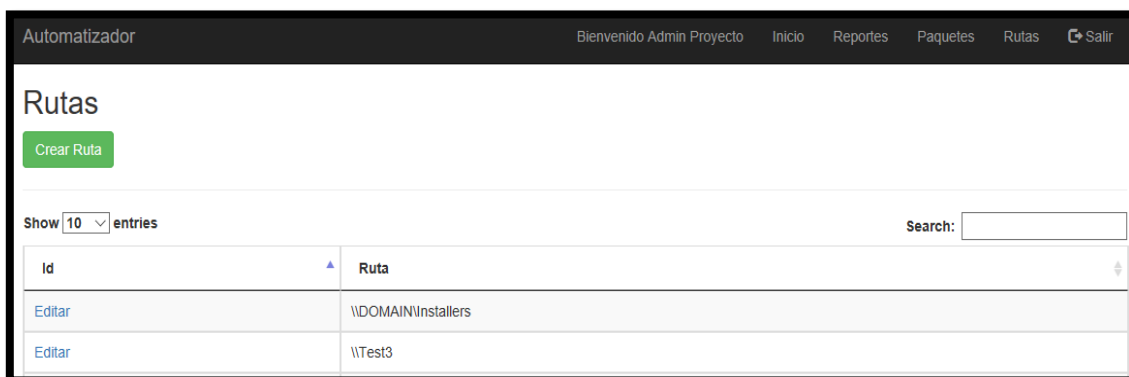


## Ilustración 17 - Perfil administrador

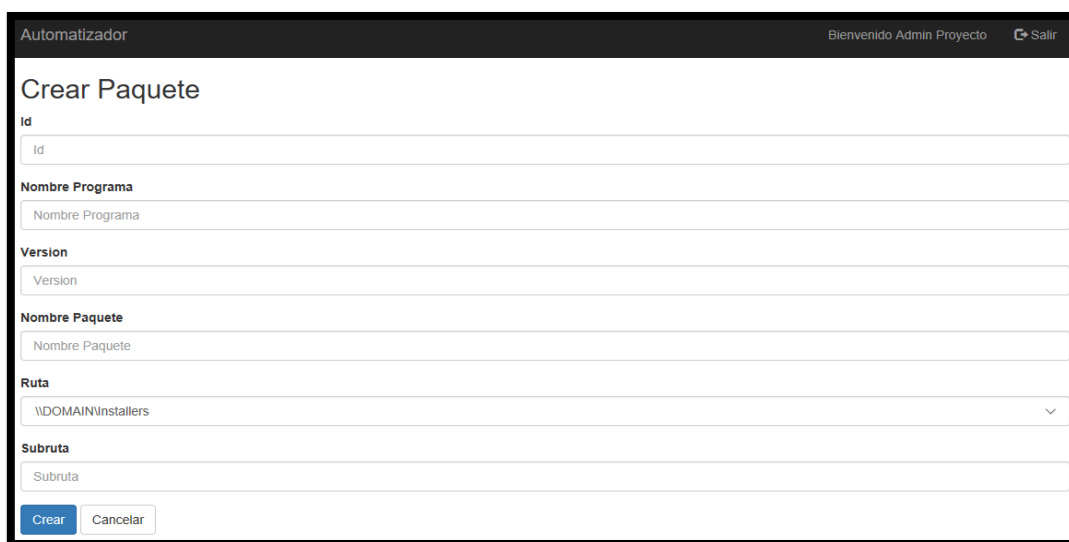


Un usuario con rol de administrador puede crear registros de paquetes automatizados (referencia de paquetes automatizados AutoIT), referenciar rutas de ubicación, conceder permisos sobre paquetes a grupos de directorio activo y acceder a reportes de instalaciones realizadas a través de la plataforma.

## Ilustración 18 - Opción Rutas



## Ilustración 19 - Opción Crear paquete

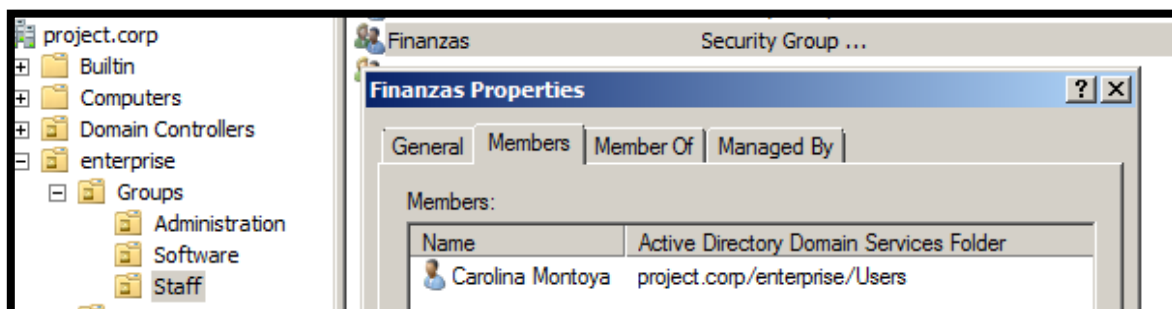


The screenshot shows a web application titled 'Automatizador'. At the top right, it says 'Bienvenido Admin Proyecto' and has a 'Salir' button. The main heading is 'Crear Paquete'. Below this, there are several input fields: 'Id' (with a sub-field 'Id'), 'Nombre Programa' (with a sub-field 'Nombre Programa'), 'Version' (with a sub-field 'Version'), 'Nombre Paquete' (with a sub-field 'Nombre Paquete'), 'Ruta' (with a sub-field '\DOMAIN\Installers' and a dropdown arrow), and 'Subruta' (with a sub-field 'Subruta'). At the bottom left, there are two buttons: 'Crear' (highlighted in blue) and 'Cancelar'.

Hasta aquí, un usuario con rol de administrador habrá podido crear los registros de paquetes de automatización para que los usuarios con distinto rol puedan acceder a la solución a verificar sus accesos a instalaciones automatizadas. Sólo basta con que el administrador del servicio de directorio activo ingrese a algún usuario final a un grupo existente, para que en pantalla pueda ver la lista de paquetes a los cuales tiene permiso.

Así mismo, en el directorio activo existen usuarios de prueba que no pertenecen al grupo PRY\_APP\_ADMINS; acmontoya es uno de ellos:

## Ilustración 20 - AD Permisos Usuario



## Ilustración 21 - Login Rol Usuario

Automatizador

### Iniciar Sesión

acmontoya

●●●●●●●●●●

Ingresar

## Ilustración 22 - Perfil Usuarios

Automatizador

Bienvenido Carolina Montoya [Salir](#)

# Bienvenido Usuario

En esta seccion puede consultar los paquetes que le son permitidos instalar

Show  entries

Search:

Nombre	Versión	Accion
Atom (AUTOIT_Atom_x64.exe)	2	<a href="#">Instalar</a>
Greenshot (greenshot.exe)	1	<a href="#">Instalar</a>

Showing 1 to 2 of 2 entries

Previous **1** Next

## 6. RESULTADOS OBTENIDOS

El usuario final podrá solicitar una instalación haciendo clic en “Instalar” sobre el registro correspondiente:

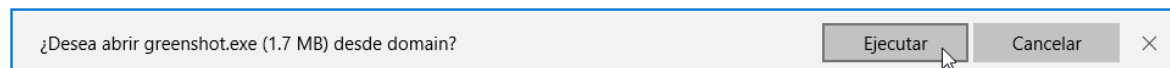
**Ilustración 23 - Función instalar software**



**Ilustración 24 - Descargando Aplicación**

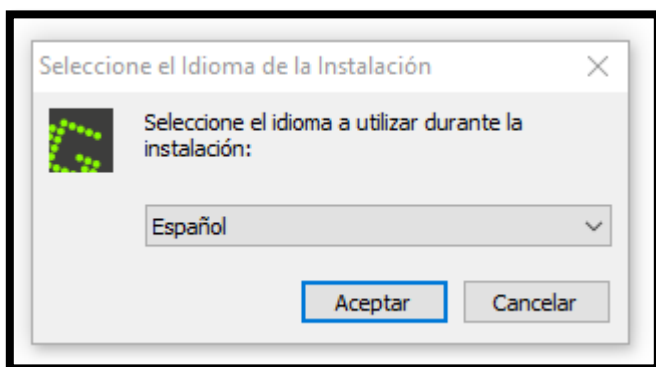


**Ilustración 25 - Ejecutar aplicación automatizada**



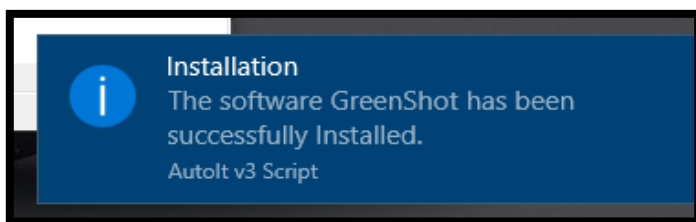
Comenzará la instalación automatizada sin ninguna acción del usuario:

## Ilustración 26 - Despliegue de aplicación



Luego el agente AutoIT correspondiente notificará cuando la instalación se encuentre realizada:

## Ilustración 27 - Notificación de instalación completa



El usuario final ahora tiene el programa instalado de acuerdo con su requerimiento a través de la solución.

El usuario administrador puede ver los paquetes de instalaciones automatizadas ejecutados en el historial:

## Ilustración 28 - Reportes de instalaciones realizadas

Automatizador


Bienvenido Admin Proyecto

Inicio

Reportes

Paquetes

Rutas

 Salir

Reportes

Show 

10

 entries

Search:

<div>Id</div>	<div>Usuario</div>	<div>Maquina</div>	<div>Fecha</div>	<div>Nombre Programa</div>
28	acmontoya	AAVG-LAPTOP	2017-11-17 12:03	Greenshot
29	acmontoya	AAVG-LAPTOP	2017-11-17 03:52	Greenshot
26	acmontoya	AAVG-LAPTOP	2017-11-16 11:57	Greenshot
27	acmontoya	AAVG-LAPTOP	2017-11-16 11:57	Greenshot
22	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:11	Atom
23	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:11	Atom
24	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:11	Atom
25	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:11	Atom
18	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:10	Atom
19	aaavengoechea	AAVG-LAPTOP	2017-11-16 11:10	Atom

Showing 1 to 10 of 29 entries

Previous

1

2

3

Next

## **7. CONCLUSIONES**

En el entorno empresarial es común encontrar procesos que no están automatizados, por lo cual es importante encontrar esos puntos faltantes a través de tecnologías que están a nuestro alcance y que entregan beneficios a la organización. Implementaciones como el despliegue automatizado de software, es una muestra de que se pueden optimizar recursos identificando este tipo de procesos; cuando las métricas se ven involucradas en la entrega de un servicio se vuelve intuitivo el modo en que se puede solucionar un problema.

El proyecto de grado ha demostrado que las oportunidades para encontrar soluciones se pueden dar en cualquier tipo de entorno, en esta ocasión se ha entregado una solución que devuelve valor a un entorno empresarial y que además al ser de código no propietario, es adaptable para otro tipo de entornos que se quieran optimizar partiendo de una idea de realizar procesos desatendidos.

Cuando se llevan a cabo proyectos por el estilo de la solución para automatizar instalaciones en ambientes Windows, es de notar que el conocimiento debe ser integral. Conforme se iban realizando los avances, el estudio y el desarrollo fue necesario aplicar conocimientos de infraestructura, redes y configuración de múltiples servicios de entorno Windows para poder simular un entorno corporativo; implícitamente fue obligatorio realizar el diseño de una infraestructura con servicios que normalmente se encuentran implementados hoy en día. Es por este tipo de retos, que participar de este tipo de proyectos amplían el umbral del conocimiento, siempre encaminando a nuevos entornos, para estar siempre en constante aprendizaje.

## **8. TRABAJOS FUTUROS**

El proyecto de automatizar instalaciones en plataformas Windows, es un proyecto que se puede extender tanto como se quiera automatizar procesos manuales. El enfoque que se le dio está ligado al despliegue de aplicaciones parametrizadas, no obstante, fue diseñada su completa arquitectura MVC y desarrollado hasta el prototipo que demostró la prueba de concepto frente a la solución final. Con el pasar del tiempo se encontró que se puede realizar un proceso más directo con el desarrollo de un protocolo, que realice un proceso de despliegue automático y más seguro. A causa de lo anterior, se hace necesaria la implementación de un agente permanente en cada máquina asociada al dominio.

Otra funcionalidad que se podría implementar a futuro es el módulo de resolución de problemas automático, en el cual un usuario requiera soporte básico sobre aplicaciones corporativas y el mismo agente se encargue de solucionarlo con un script ejecutándose en segundo plano.

Como mejora a la solución, se puede implementar la funcionalidad de instalación en modo silencioso o en segundo plano, para que el usuario final tenga la percepción de una ejecución más limpia y sin interrupciones.

## 9. BIBLIOGRAFIA

- Allen, R., & Hunter, L. E. (2006). *Active directory cookbook*. O'Reilly. Retrieved from [https://books.google.es/books?hl=es&lr=&id=NnJHAgAAQBAJ&oi=fnd&pg=PR5&dq=active+directory&ots=sJbpqyAlVI&sig=Hxf7j\\_9VrwMVkm59Th-nekrOZGI#v=onepage&q=active+directory&f=false](https://books.google.es/books?hl=es&lr=&id=NnJHAgAAQBAJ&oi=fnd&pg=PR5&dq=active+directory&ots=sJbpqyAlVI&sig=Hxf7j_9VrwMVkm59Th-nekrOZGI#v=onepage&q=active+directory&f=false)
- AutoIT, S. (n.d.). AutoIT Script Editor. Retrieved from <https://www.autoitscript.com/site/autoit-script-editor/>
- Bautista, J. (1999). Programación Extrema ( Xp ), 7; creacion:2013;Recuperado: 1 febrero 2016. Retrieved from [http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACIÓN EXTREMA.pdf](http://ingenieriadesoftware.mex.tl/images/18149/PROGRAMACIÓN%20EXTREMA.pdf)
- Bolosky, W. J., Barrera, J. S., Draves, R. P., Fitzgerald, R. P., Gibson, G. A., Jones, M. B., ... Rashid, R. F. (1996). The Tiger Video Fileserver. Retrieved from <https://pdfs.semanticscholar.org/69dd/50fc94560be5b1cc0d3b28e9d2ce772a9fa9.pdf>
- Bonnet, N. (2014). *Windows Server 2012 R2: las bases imprescindibles para administrar y configurar su servidor*. ENI. Retrieved from <https://books.google.es/books?hl=es&lr=&id=464Hd3t0CsIC&oi=fnd&pg=PA13&dq=iis+servidor+windows&ots=9RCIYbDB2J&sig=UADeVDJmyCGliHXIfZj1eVWGeQU#v=onepage&q=iis+servidor+windows&f=false>
- Brand, J., & Balvanz, J. (2005). Automation is a breeze with Autolt. In *Proceedings of the 33rd annual ACM SIGUCCS conference on User services - SIGUCCS '05* (pp. 12–15). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1099435.1099439>
- Capeáns Hurtado, C. A., & Rodríguez Puente, R. (2015). Compatibilidad tecnológica en el despliegue de sistemas de gestión empresarial. *Compatibility Technology in of Enterprise Systems Implementation.*, 9(2), 29–51. Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=fua&AN=102611774&lang=es&site=ehost-live>
- CAPITULO III. (n.d.). Retrieved from <http://repositorio.utc.edu.ec/bitstream/27000/1845/3/T-UTC-1336%283%29.pdf>
- Carlos, J., Rojas, V., Ariel, J., & Alegría, H. (2007). Procesos Software Software.
- Casas González, J. C. (2008). Diseño y desarrollo de una aplicación cliente-servidor usando como método de transferencia de datos clientes de correo electrónico. Retrieved from <http://repositorio.upct.es/handle/10317/749#?>



- Cataldi, Z., & Salgueiro, F. (2007). Software libre y código abierto en educación. *Quaderns Digitals*, 48, 1–12. Retrieved from [http://www.quadernsdigitals.net/datos/hemeroteca/r\\_1/nr\\_751/a\\_10208/10208.pdf](http://www.quadernsdigitals.net/datos/hemeroteca/r_1/nr_751/a_10208/10208.pdf)
- Center, I. K. (2006). XDR - IBM. Retrieved from [https://www.ibm.com/support/knowledgecenter/es/ssw\\_aix\\_61/com.ibm.aix.net.workcomm/nfs\\_xdr\\_protocol.htm](https://www.ibm.com/support/knowledgecenter/es/ssw_aix_61/com.ibm.aix.net.workcomm/nfs_xdr_protocol.htm)
- Chadwick, D. (2005). Threat Modelling for Active Directory. In *Communications and Multimedia Security* (pp. 173–182). New York: Springer-Verlag. [https://doi.org/10.1007/0-387-24486-7\\_13](https://doi.org/10.1007/0-387-24486-7_13)
- Charles J. Gazdik, Kelli Kinkela, D. M. (1998). Software installation process using abstract data and program files. Retrieved from <https://www.google.com/patents/US6301708>
- Chinchilla-Arley, R. (2011). El software libre: Una alternativa para automatizar unidades de información. *Bibliotecas: Revista de La Escuela de Bibliotecología, Documentación E Información*, 29(1), 7--24. Retrieved from <http://www.revistas.una.ac.cr/index.php/bibliotecas/article/view/1557>
- Corporation, M. (n.d.). ¿Qué es System Center Configuration Manager? – IT PRO Colombia. Retrieved September 26, 2017, from <https://blogs.technet.microsoft.com/itprocol/2012/01/02/qu-es-system-center-configuration-manager/>
- Corporation, M. (2012). SMB Microsoft. Retrieved from [https://msdn.microsoft.com/es-es/library/hh831795\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831795(v=ws.11).aspx)
- Desmond, B. (2008). *Active Directory: Designing, Deploying, and Running Active Directory*. O'Reilly Media, Inc. Retrieved from [https://books.google.es/books?hl=es&lr=&id=exlzxcsE-7QC&oi=fnd&pg=PR5&dq=active+directory&ots=JGW7E0Gil\\_&sig=\\_HZzForWaVWtZLaFP9inDahTeEA#v=onepage&q=active directory&f=false](https://books.google.es/books?hl=es&lr=&id=exlzxcsE-7QC&oi=fnd&pg=PR5&dq=active+directory&ots=JGW7E0Gil_&sig=_HZzForWaVWtZLaFP9inDahTeEA#v=onepage&q=active directory&f=false)
- Dias, J. (2002). A Guide to Microsoft Active Directory ( AD ) Design. *Scenario*. Retrieved from [http://programando.ucoz.com/\\_ld/0/44\\_MS\\_Active\\_Direc.pdf](http://programando.ucoz.com/_ld/0/44_MS_Active_Direc.pdf)
- Flesner, A. (2007). *Autolt v3: your quick guide*. O'Reilly. Retrieved from <https://books.google.es/books?hl=es&lr=&id=x8Nw7VPwjl4C&oi=fnd&pg=PA10&dq=autoit&ots=3gPIQyT0R5&sig=EqLqQ3pgsFDw5wYG9NYchb7aVqw#v=onepage&q=autoit&f=false>
- González, M. M. A., Santos, B. S. N., Vargas, A. R., Martín-Gutiérrez, J., & Orihuela, A. R. (2013). Virtual worlds. Opportunities and challenges in the 21st century. *Procedia Computer Science*, 25(6), 330–337. <https://doi.org/10.1016/j.procs.2013.11.039>
- Guzman, I. A., Davila, T., Garcia, G., & Caicedo Rossi, G. A. (1998). Diseño bajo la arquitectura cliente-servidor del sistema integrado de notas. Retrieved from

- <http://www.dspace.espol.edu.ec/handle/123456789/3283#?>
- IBM. (n.d.). IBM NetServer/QNTC and SMB (Server Message Block) Version 2.0 and Version 3.0 - España. Retrieved August 30, 2017, from <http://www-01.ibm.com/support/docview.wss?uid=nas8N1011878>
- Instituto Nacional de Tecnologías Educativas y Formación del Profesorado. (2017). NFS-RecursosTIC WEB. Nfs. Retrieved from <http://recursostic.educacion.es/observatorio/web/gl/software/software-general/733-nfs-sistema-de-archivos-de-red>
- José H. Canós, M. C. P. P. L. (2012). Metodologías Ágiles en el Desarrollo de Software. Retrieved from <http://roa.ult.edu.cu/handle/123456789/476>
- Joskowicz, I. J. (2008). Reglas y Prácticas en eXtreme Programming. Retrieved from <http://iie.fing.edu.uy/~josej/docs/XP - Jose Joskowicz.pdf>
- Kasiak, T., & Godoy, D. A. (2012). Simulación de proyectos de Software desarrollados con XP: Subsistema de desarrollo de tareas. Retrieved from <http://sedici.unlp.edu.ar/handle/10915/18976>
- Kim Douglas Mason, V. L. (2006). Software installation process. Retrieved from <https://www.google.com/patents/US8607059>
- Kralik, L. (n.d.). Analysis for Automated Unattended Installation, 163–167.
- Macías Sánchez, M. (n.d.). Gestión del cambio y planificación estratégica. Retrieved from [http://www.palermo.edu/economicas/cbrs/pdf/pbr12/PBR\\_13\\_04.pdf](http://www.palermo.edu/economicas/cbrs/pdf/pbr12/PBR_13_04.pdf)
- manageengine. (2017). Automatización de Software. Retrieved from <http://manageengine.mx/desktop-mdm/automatico-el-despliegue-de-software-desde-una-localidad-central.html>
- Manzoor, U., & Nefti, S. (2010). QUIET: A methodology for autonomous software deployment using mobile agents. *Journal of Network and Computer Applications*, 33(6), 696–706. <https://doi.org/10.1016/j.jnca.2010.03.015>
- McGuire, I. T. D., Sliger, M. V, Welch, D. C., Us, W. A., Aul, G. J., Us, W. A., ... Alan, B. (2002). METHOD AND SYSTEM FOR DOWNLOADING UPDATES FOR SOFTWARE INSTALLATION. *McGuire*, 1(12).
- Mejías, Cintado, S. (n.d.). “CÓMO CONFIGURAR INTERNET INFORMATION SERVER” Retrieved from [http://s219540635.mialojamiento.es/feb2010/CONFIGURAR\\_INFORMATION\\_SERVER.pdf](http://s219540635.mialojamiento.es/feb2010/CONFIGURAR_INFORMATION_SERVER.pdf)
- Microsoft. (2016). Introducción al servidor web (IIS). Retrieved August 30, 2017, from [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx)
- Microsoft corp. (n.d.). FileServer Microsoft. Retrieved September 11, 2017, from [https://msdn.microsoft.com/es-es/library/hh831487\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831487(v=ws.11).aspx)

- Minera, F. (2008). *Curso de Programación PHP: aprenda a programar con el lenguaje más flexible*. Gradi.
- Profesorado, I. N. de T. E. y F. del. (2006). NTFS RecursosTICS. Retrieved from <http://recursostic.educacion.es/observatorio/web/gl/software/software-general/733-nfs-sistema-de-archivos-de-red>
- Reig, D. (2010). ¿Qué son los servidores web y por qué son necesarios? Retrieved August 29, 2017, from <https://duplika.com/blog/que-son-los-servidores-web-y-por-que-son-necesarios/>
- Rosa, F., & Heinz, F. (2007). *Software Libre Guía Práctica sobre*.
- SCITE. (n.d.). Autolt. Retrieved September 11, 2017, from <https://www.autoitscript.com/autoit3/docs/>
- SoftinventiveLab. (2017). Total Software Deployment: implementación e instalación de software remoto automatizado para su red corporativa. Retrieved August 21, 2017, from <https://www.softinventive.es/total-software-deployment/>
- Stallman, R. (2004). *Software libre para una sociedad libre*. Retrieved from <http://libros.metabiblioteca.org/bitstream/001/144/8/84-933555-1-8.pdf>
- Thibaud, C. (2006). *MySQL 5: instalación, implementación, administración, programación*. ENI Editions. Retrieved from <https://books.google.es/books?hl=es&lr=&id=wY0bHPmW-NUC&oi=fnd&pg=PA6&dq=mysql+database+server+scripting&ots=UkoIBPhFqR&sig=PCOYAIFDp1o40FmFHZdO8QOI0PU#v=onepage&q&f=false>
- Tsirogiannis, D., Harizopoulos, S., & Shah, M. A. (2010). Analyzing the energy efficiency of a database server. In *Proceedings of the 2010 international conference on Management of data - SIGMOD '10* (p. 231). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1807167.1807194>
- Wang, J.-A., & Liu, K. C. (2003). Fracture Toughness Determination Using Spiral-Grooved Cylindrical Specimen and Pure Torsional Loading. *United States Patent Application Publication, 2003002432(A1)*, 1–14. <https://doi.org/US2010/0311130A1>
- Wang, Y., & Rowe, L. A. (n.d.). Cache Consistency and Concurrency Control in a Client/Server DBMS Architecture. Retrieved from <https://www.cc.gatech.edu/computing/Database/readinggroup/articles/p367-wang.pdf>
- Yikang Xu, Uresh K. Vahalia, Xiaoye Jiang, Uday Gupta, P. T. (1999). File server system using file system storage, data movers, and an exchange of meta data among data movers for file locking and direct access to shared file systems. Retrieved from <https://www.google.com/patents/US6324581>

## GLOSARIO

A continuación, se relaciona una lista de conceptos claves al momento de leer el presente proyecto:<sup>1</sup>

**Ambiente:** Todo aquello que está afuera de los límites del sistema se considera como entorno o ambiente del mismo.

**Automatización:** Conversión de toda actividad repetitiva para que se desarrollen por sí solas y sin la participación directa de un individuo

**Despliegue:** Proceso por el que una aplicación informática pasa a estar lista para utilizarse.

**Directorio Activo:** Servicio de directorio en una red distribuida de computadores implementada por Microsoft

**Dominio:** Conjunto de computadoras conectadas en una red informática que confían a uno de los equipos de dicha red, la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red

**Entorno:** Sinónimo de ambiente (Véase Ambiente).

**File Server:** Tipo de servidor que almacena y distribuye diferentes tipos de archivos informáticos entre los clientes de una red de computadoras

**HTTP:** Hypertext Transfer Protocol o protocolo de transferencia de hipertextos

**LAN:** Local Area Network o red de área local

**LDAP:** Lightweight Directory Access Protocol o Protocolo ligero de acceso a directorios

**Plataforma:** Potente herramienta de gestión empresarial conformada por un conjunto de hardware

**Protocolo:** Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red

**Red:** Conjunto de dispositivos interconectados entre sí a través de un medio, que intercambian información y comparten recursos.

**Scripts:** Lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador.

**Servicios:** Conjunto de actividades que buscan responder a las necesidades de un cliente por medio de un cambio de condición en los bienes informáticos

**Servidor web:** Programa informático que procesa una aplicación del lado del servidor que provee el servicio.

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas

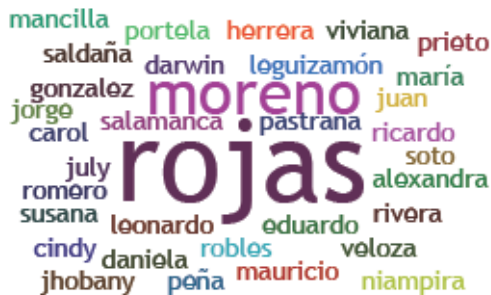
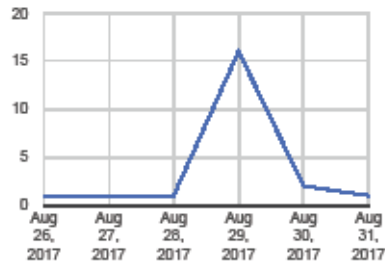
## ANEXO A - Encuesta

17/11/2017

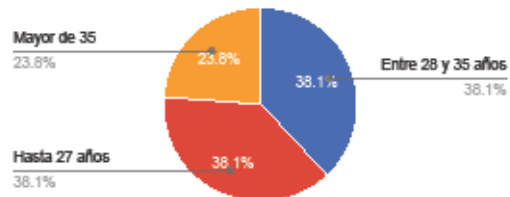
copy of [Addon Advanced summary] Encuesta gestión de instalaciones - Wed Oct 25 2017

21

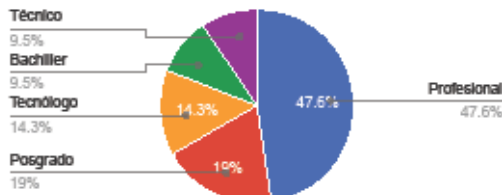
Marca temporal NOT\_FOUND



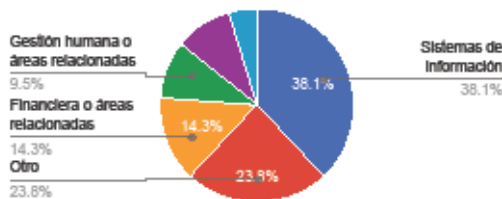
Edad



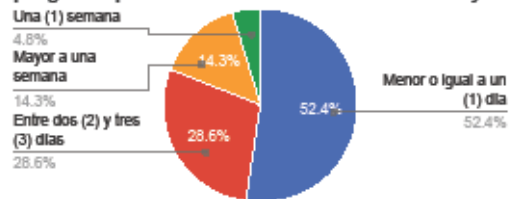
Grado de estudios alcanzados



Nombre de sección o área



Cuando requiere de la instalación de un programa para el desarrollo de sus funciones y...



<https://awesome-fable.com/-KzCNsGVWCibxxemxSVI/print>

1/2

**ANEXO B – Documento de especificación de requerimientos.**



**SOLUCIÓN INSTALACIONES AUTOMATIZADAS SOBRE SISTEMAS  
OPERATIVOS WINDOWS**

**DARWIN ARNULFO LEGUIZAMÓN QUINCHE CÓDIGO: 625091**

**ALEXANDER ARTURO VENGOECHEA GUERRERO CÓDIGO: 625084**

**ESPECIFICACIÓN DE REQUERIMIENTOS**

**DIEGO ALBERTO RINCÓN YÁÑEZ MCS**

Contenido	
Control de Cambios.....	65
Introducción.....	66
Resumen.....	66
Propósito.....	66
Autores.....	66
Ámbito del Sistema .....	66
Definiciones, Acrónimos y Abreviaturas.....	67
Visión General del Documento .....	67
Alcance .....	68
Descripción.....	68
Funciones del producto.....	68
Características de los Usuarios .....	69
Restricciones .....	70
Suposiciones y Dependencias .....	70
Requisitos futuros .....	70
Metodología .....	70
Requerimientos Específicos.....	71
Funcionalidad.....	71
Interfaces.....	75
De sistema .....	75
De usuario.....	75
De software.....	75
Requerimientos de Licenciamiento .....	75
Glosario.....	76
Bibliografía .....	77



## Control de Cambios

Fecha	Autor	Versión	Comentarios
11 de octubre de 2017	Alexander Vengoechea / Darwin Leguizamón	1.0	Reconocimiento de documento e identificación de requerimientos generales
24 de octubre de 2017	Alexander Vengoechea / Darwin Leguizamón	2.0	Concepción de requerimientos y contenidos del documento detallados
15 de noviembre de 2017	Alexander Vengoechea / Darwin Leguizamón	3.0	Revisión de conceptos y glosario

## Introducción

Este documento es una Especificación de Requisitos Software (ERS) del sistema o solución de automatización de instalaciones en ambientes Windows.

## Resumen

Se exponen los requerimientos identificados para darles solución posteriormente a través de una herramienta que satisfaga las necesidades identificando usuarios objetivo.

## Propósito

El presente documento tiene como razón principal exponer las especificaciones funcionales y no funcionales de requisitos de software (ERS) para la plataforma de automatización de instalaciones en ambientes Windows. El propósito es definir los requisitos de forma clara para concebir un sistema o solución que satisfaga las necesidades identificadas.

Así mismo, el presente documento principalmente está dirigido a arquitectos y desarrolladores de software, quienes tendrán la visión y el alcance de la plataforma mediante los requerimientos expuestos.

## Autores

Nombre	Darwin Leguizamón
Rol	Analista y programador
Categoría Profesional	Análisis metodológico y desarrollo
Contacto	<b><u><a href="mailto:daleguizamon91@ucatolica.edu.co">daleguizamon91@ucatolica.edu.co</a></u></b>

Nombre	Alexander Vengoechea
Rol	Analista y programador
Categoría Profesional	Análisis metodológico y desarrollo
Contacto	<b><u><a href="mailto:aavengoechea84@ucatolica.edu.co">aavengoechea84@ucatolica.edu.co</a></u></b>

## Ámbito del Sistema

Se requiere una plataforma capaz de controlar a través de permisos de acceso, la ejecución de paquetes de automatización a usuarios finales. Así mismo, permitirá consultar registros de instalaciones realizadas a través de la plataforma. Lo anterior se da en un ambiente Windows, cuyos servicios son de directorio activo, motor de base de datos y servidor de archivos.

## Definiciones, Acrónimos y Abreviaturas

Nombre	Descripción
AD	Active Directory o Directorio Activo por su traducción al español. Es un servicio de repositorio en entorno Windows que contiene información de objetos de red tales como grupos de usuarios y usuarios (Desmond, 2008)
Base de datos	Servicio contenedor de registros en un sistema computarizado sobre el cual posteriormente se pueden realizar operaciones de consultas (Date & Ruiz Faudón, 2001)
DFS	Distributed File System o Sistema de Distribución de Archivos. Servicio de servidor de archivos que retorna uno o más archivos desde un espacio de nombres definido en una red ("Distributed file system providing a unified name space with efficient name resolution," 1993)
ERS	Especificación de Requisitos de Software
HTTP	Hyper Text Transfer Protocol o Protocolo de Transferencia de Hypertexto. Protocolo a nivel de aplicación para transmisión de archivos de hipermedia (Fielding et al., 1999)
IIS	Internet Information Services o Servicio de Información de Internet. Servicio Microsoft para publicar sitios con tecnología web
LDAP	Protocolo de acceso a ligero a directorio utilizado por el servicio de directorio activo (Donnelly, n.d.)
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional

## Visión General del Documento

En el documento se realiza una introducción en cuanto a requerimientos del sistema. Partiendo de una necesidad identificada global, se desglosan dichos requerimientos que dan lugar a la conformación y configuración de la plataforma para que la necesidad sea satisfactoriamente suplida.

Finalmente se presentan las conclusiones del resultado del análisis de los requerimientos tanto funcionales como no funcionales.

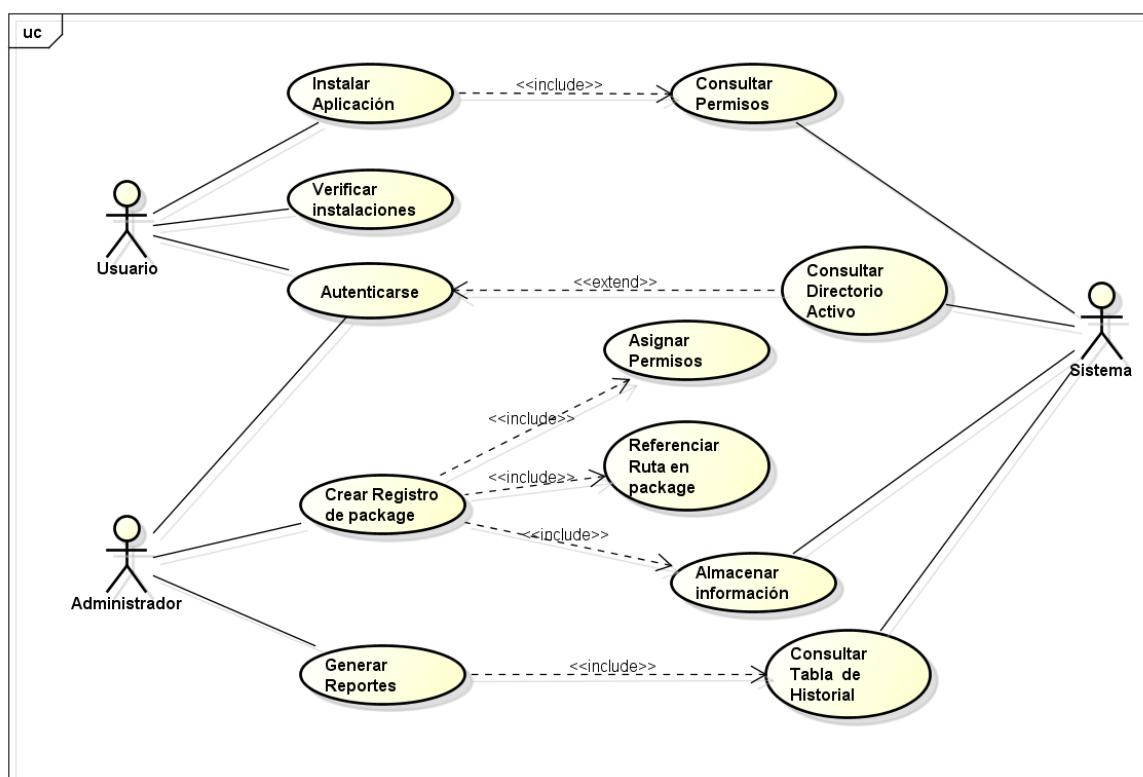
### Alcance

La solución, componentes y servicios propuestos incluyen configuraciones principalmente diseñados para ambientes Microsoft Windows sobre el cual se desarrolla el proyecto. Con esto se pretende dar una solución adecuada al tema de instalaciones automatizadas bajo las condiciones propuestas en el proyecto de grado acerca de la situación actual en muchas empresas de mediano y gran tamaño.

## Descripción

### Funciones del producto

Ilustración 29 - Casos de uso



Los usuarios del sistema identificados usuario final y usuario administrador, son quienes utilizan la plataforma con acciones de autenticación para una posterior ejecución de acciones a las cuales tienen permisos sobre el sistema. Existe una integración entre el sistema y los servicios existentes sobre ambientes Windows permitiendo validar la información pertinente de acuerdo con el usuario identificado en la autenticación.

El usuario final podrá:

- Autenticarse: la autenticación define el rol del usuario dentro del sistema para las acciones posteriores.
- Instalar una aplicación: realizar una instalación sobre la cual cuenta con permisos a través de un automatizador.
- Verificar instalaciones: a través de un automatizador podrá verificar si el programa que está intentando instalar ya existe en el equipo de cómputo actual.

El usuario administrador podrá:

- Autenticarse: la autenticación define el rol del usuario dentro del sistema para las acciones posteriores.
- Crear registro de package: a través de esta acción, el usuario administrador podrá crear registros de paquetes automatizados para su posterior uso por parte de los usuarios finales.
- Asignar permisos: como consecuencia de registrar un nuevo paquete en el sistema, al mismo se le asignarán permisos sobre un grupo del servicio de directorio activo.
- Referenciar ruta en package: se podrá vincular una ruta del servicio de servidor de archivos en donde se encuentra el paquete automatizado.
- Almacenar información: como consecuencia de la creación del registro de un paquete en el formulario adecuado, podrá dar la orden al sistema de almacenar el registro en el servicio de base de datos.
- Generar reportes: acceder a reportes de programas instalados a través de la plataforma.
- Consultar tabla de historial: al intentar generar un reporte, primero se consultará la tabla de historial sobre la base de datos de la solución.

El sistema deberá ser capaz de:

- Consultar permisos: en grupos del servicio de directorio activo, el sistema consultará la pertenencia del usuario para luego redirigirlo en el sistema al formulario adecuado.
- Consultar directorio activo: acción consecuente a consultar permisos ya sean de paquetes automatizados o de acciones de autenticación.
- Almacenar información: ejecutar las acciones de guardado de registros una vez le sea dada la orden a través de la creación de un registro de paquete realizada por un usuario administrador de la plataforma.
- Consultar tabla de historial: ejecutar dicha consulta consecuencia de la petición de generación de reporte por parte de un usuario administrador.
- 

### Características de los Usuarios

En la solución se identifican los siguientes roles de usuarios:

- Usuario final: toda aquella persona que hace uso del sistema, el cual le permite realizar una instalación desatendida.
- Usuario administrador: toda aquella persona que preparará los registros del sistema para que el rol de usuario final pueda utilizar la plataforma correctamente. Este rol puede establecer permisos sobre paquetes de instalación y generar reportes.

#### Restricciones

- La solución será probada bajo un ambiente de servicios Windows.
- Lenguaje de programación: C# - Visual Studio .Net
- El establecimiento de permisos se realizará en integración con directorio activo o AD.
- La solución será probada en ambiente virtualizado simulando una red física de servicios y cliente.
- La plataforma será usada como solución web.

#### Suposiciones y Dependencias

##### Suposiciones:

- Los requerimientos de la solución no cambiarán durante el desarrollo de la solución de la plataforma.
- Se asegurará la disponibilidad de los diferentes servicios al momento de realizar una prueba de la solución.

##### Dependencias:

- Disponibilidad de los equipos de desarrollo y líderes del proyecto.
- Conocimiento adecuado para instalar y compartir los servicios pertinentes para el funcionamiento de la plataforma.

#### Requisitos futuros

- Implementar alta disponibilidad en los servicios correspondientes al proyecto.

#### Metodología

- Una vez identificados los requerimientos, se establecerán los servicios base que apoyarán la construcción del prototipo.
- El proyecto de construcción del prototipo de la solución será elaborado con la metodología ágil XP.

## Requerimientos Específicos

### Funcionalidad

A continuación, se detallan los requisitos que el sistema deberá implementar en funcionales (RF) y No Funcionales (RNF):

Código	RF01
Nombre de requerimiento	Ingreso al sistema
Prioridad	Alta
Descripción	Para el proceso de ingreso se debe preguntar por usuario y contraseña para la correcta identificación de rol de usuario
Entradas	Acceder a la URL del aplicativo
Proceso:	La plataforma debe exponer un formulario de autenticación a través del cual se identifiquen los roles de usuarios
Salidas:	Redireccionamiento a la interfaz adecuada de acuerdo con el rol del usuario

Código	RF02
Nombre de requerimiento	Consultar permisos asignados
Prioridad	Alta
Descripción	La plataforma debe consultar permisos asignados a un usuario sobre los paquetes de automatización para exponer al mismo los paquetes permitidos
Entradas	Acceder a la plataforma con rol de usuario final (RF01)
Proceso:	La plataforma deberá consultar en base de datos los registros en los cuales se encuentra referenciado un grupo de directorio activo que consecuentemente tiene al usuario configurado
Salidas:	Ejecutar el/los paquetes de automatización permitidos

Código	RF03
Nombre de requerimiento	Ejecutar paquetes de automatización
Prioridad	Alta
Descripción	La plataforma deberá ser capaz de ejecutar un paquete de automatización en un servidor de archivos
Entradas	Validación de permisos asignados (RF02)
Proceso:	Una vez validados los permisos sobre un registro de paquete de automatización, la plataforma deberá ejecutar dicho paquete sobre la máquina local del

	usuario final
Salidas:	Instalar programa

Código	RF04
Nombre de requerimiento	Crear registro de un paquete de automatización
Prioridad	Alta
Descripción	La plataforma debe guardar registros que relacionen el instalador automatizado con grupos de directorio activo a los cuales pertenecen usuarios finales
Entradas	Acceder a la plataforma con rol de usuario administrador (RF01)
Proceso:	La plataforma guardará los registros en base de datos para ser consultados posteriormente por usuarios finales
Salidas:	Guardado del registro

Código	RF05
Nombre de requerimiento	Leer grupos de directorio activo
Prioridad	Alta
Descripción	La plataforma debe ser capaz de leer los grupos de directorio activo a los que pertenece un usuario final. Así mismo, desde el rol del usuario administrador, la plataforma debe ser capaz de devolver grupos de directorio activo sobre una OU determinada
Entradas	Autenticación con rol de usuario final o autenticación como usuario administrador (para listar grupos y dar permisos sobre el paquete) (RF01)
Proceso:	La plataforma se conectará al directorio activo realizando una consulta de pertenencia a grupos de usuarios finales autenticados en el sistema y los grupos listados en una OU específica (este último desde el rol administrador)
Salidas:	Listar paquetes a los cuales tiene permiso el usuario

Código	RF06
Nombre de requerimiento	Generar reportes de instalaciones
Prioridad	Alta
Descripción	La plataforma debe consultar la información del historial de los usuarios finales que ejecutan bajo demanda los paquetes de instalación a través de los registros del aplicativo. Debe poder generar



	reportes de instalaciones realizadas a través de la plataforma por usuario y por registro de paquete de instalación
Entradas	Autenticación con rol de usuario administrador (RF01)
Proceso:	La plataforma consultará una tabla de historial en la base de datos y filtrará la consulta de acuerdo con el reporte solicitado por el usuario administrador
Salidas:	Generar reporte para visualización en pantalla

Código	RF07
Nombre de requerimiento	Verificar instalación en equipo local.
Prioridad	Alta
Descripción	La plataforma debe verificar si la instalación de un programa solicitado por medio de un paquete de automatización ya se encuentra realizada en la máquina local
Entradas	Ejecución del paquete automatizado (RF03)
Proceso:	El script automatizado será capaz de realizar la verificación si un programa está instalado o no
Salidas:	Si el programa no está instalado, el script procede a la instalación

Código	RNF01
Nombre de requerimiento	Interfaz gráfica
Prioridad	Alta
Descripción	La interfaz gráfica de la plataforma debe ser sencilla y entendible de acuerdo con las funciones
Entradas	Ninguna
Proceso:	Se realizarán formularios sencillos y agradables visualmente
Salidas:	Uso por parte del usuario

Código	RNF02
Nombre de requerimiento	Usabilidad
Prioridad	Alta
Descripción	El aplicativo web de la plataforma debe ser sencillo, intuitivo y agradable para los usuarios
Entradas	Ninguna
Proceso:	La presentación de los formularios se realizará a modo de asistente con las acciones definidas de acuerdo con el rol del usuario
Salidas:	Uso por parte del usuario

Código	RNF03
Nombre de requerimiento	Disponibilidad
Prioridad	Alta
Descripción	La aplicación puede estar soportada por configuración de servicios sin alta disponibilidad
Entradas	Alistamiento de servidores virtuales
Proceso:	Se configurarán servicios únicos sin alta disponibilidad
Salidas:	Asegurar disponibilidad del servicio ya configurado

Código	RNF04
Nombre de requerimiento	Desempeño
Prioridad	Alta
Descripción	La plataforma deberá garantizar una respuesta rápida ante cualquier solicitud
Entradas	Configuración de máquinas virtuales e instalación y configuración de los servicios necesarios (servidor de archivos, directorio activo, base de datos y servidor web)
Proceso:	Se configurarán los servicios en máquinas virtuales simulando la colocación de los mismos dentro de una misma subred
Salidas:	Tiempo de respuesta adecuado

Código	RNF05
Nombre de requerimiento	Seguridad
Prioridad	Alta
Descripción	La plataforma deberá garantizar la seguridad y confirmación de los datos de autenticación
Entradas	Ninguna
Proceso:	Se confirmarán los datos de inicio de sesión y configuración de permisos a paquetes de instalación mediante grupos de directorio activo
Salidas:	Ninguna

Código	RNF06
Nombre de requerimiento	Soporte
Prioridad	Alta
Descripción	Se deberá describir la configuración de los servicios que intervienen en la plataforma
Entradas	Ninguna
Proceso:	Referencias a manuales de configuración de los servicios

	y configuraciones de los recursos dentro de dichos servicios para la operación de la plataforma
Salidas:	Ninguna

## Interfaces

### De sistema

Dichas interfaces corresponden a la interacción con sistemas externos para el aprovechamiento del entorno, tales como:

- Conexión a directorio activo a través de protocolo LDAP para validación de usuarios.
- Conexión a base de datos a través de sockets para consultas y guardado de información.
- Acceso a la aplicación web publicada a través de IIS mediante protocolo http.
- Acceso a servicios de servidor de archivos mediante configuración DFS en servidor Windows.
- Ejecución de script de automatización en máquina local del usuario mediante permisos de usuario de servicio configurado en directorio activo.

### De usuario

Corresponden a todas las que a través de las cuales cualquier tipo de usuario de los identificados, puede interactuar con el sistema incluyendo mouse, teclado, pantalla e interfaz gráfica a través de un navegador web.

### De software

Nombre	Descripción	Propósito
Sistema Operativo	Windows XP	Sistema operativo de versión mínima del lado cliente para acceder a la aplicación y para la ejecución de una instalación automatizada
.Net Framework 3.5	Framework de Microsoft	Versión mínima de la interfaz de ejecución de las órdenes del aplicativo web

## Requerimientos de Licenciamiento

Para la instalación de la solución, se requiere contar con un entorno de servicios Windows en red tales como servicio de directorio activo, base de datos, servicio DNS, servidor de archivos y servidor IIS.

## Glosario

**Ambiente:** Todo aquello que está afuera de los límites del sistema se considera como entorno o ambiente del mismo.

**Entorno:** Sinónimo de ambiente (Véase Ambiente).

**Protocolo:** Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red.

**Red:** Conjunto de dispositivos interconectados entre sí a través de un medio, que intercambian información y comparten recursos.

**Servicios:** Conjunto de actividades que buscan responder a las necesidades de un cliente por medio de un cambio de condición en los bienes informáticos.

**Servidor de archivos:** Tipo de servidor que almacena y distribuye diferentes tipos de archivos informáticos entre los clientes de una red de computadoras.

**Servidor web:** Programa informático que procesa una aplicación del lado del servidor que provee el servicio.

## Bibliografía

- Date, C. J., & Ruiz Faudón, S. L. M. (2001). *Introducción a los sistemas de bases de datos*. Pearson Educación. Retrieved from [https://books.google.es/books?hl=en&lr=&id=Vhum351TK8C&oi=fnd&pg=PR17&dq=definición+de+base+de+datos+sql&ots=fzH3UYe8bg&sig=tr6VwEJsNNLYPOysAAefalOfSho#v=onepage&q=definición de base de datos sql&f=false](https://books.google.es/books?hl=en&lr=&id=Vhum351TK8C&oi=fnd&pg=PR17&dq=definición+de+base+de+datos+sql&ots=fzH3UYe8bg&sig=tr6VwEJsNNLYPOysAAefalOfSho#v=onepage&q=definición+de+base+de+datos+sql&f=false)
- Desmond, B. (2008). *Active Directory : Designing, Deploying, and Running Active Directory*. O'Reilly Media, Inc. Retrieved from [https://books.google.es/books?hl=en&lr=&id=exlzxcSE-7QC&oi=fnd&pg=PR5&dq=what+is+active+directory+windows&ots=JGX2L1HrN3&sig=Vx-C8qmbTQFiwmsjLBeUTsP89fk#v=onepage&q=what is active directory windows&f=false](https://books.google.es/books?hl=en&lr=&id=exlzxcSE-7QC&oi=fnd&pg=PR5&dq=what+is+active+directory+windows&ots=JGX2L1HrN3&sig=Vx-C8qmbTQFiwmsjLBeUTsP89fk#v=onepage&q=what+is+active+directory+windows&f=false)
- Distributed file system providing a unified name space with efficient name resolution. (1993). Retrieved from <https://www.google.com/patents/US5701462>
- Donnelly, M. (n.d.). Una Introducción a LDAP. Retrieved from [http://redeslinux.com/manuales/openldap/introduccion\\_ldap.pdf](http://redeslinux.com/manuales/openldap/introduccion_ldap.pdf)
- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). RFC2616 - Hypertext transfer protocol–HTTP/1.1. *Internet Engineering Task Force*, 1–114. <https://doi.org/http://www.ietf.org/rfc/rfc2616.txt>

## **ANEXO C – Documento de diseño**



### **SOLUCIÓN INSTALACIONES AUTOMATIZADAS SOBRE SISTEMAS OPERATIVOS WINDOWS**

**DARWIN ARNULFO LEGUIZAMÓN QUINCHE      CÓDIGO: 625091**

**ALEXANDER ARTURO VENGOECHEA GUERRERO      CÓDIGO: 625084**

### **DOCUMENTO DE DISEÑO DE SOFTWARE**

**DIEGO ALBERTO RINCÓN YÁÑEZ MCS<sub>c</sub>**

Contenido	
Control de Cambios.....	80
Introducción.....	81
Alcance .....	81
Supuestos .....	81
Dependencias .....	81
Restricciones .....	82
Metodología de Pruebas .....	82
Arquitectura del Sistema .....	84
Nivel General .....	84
Diagrama de Componentes .....	85
Diagrama de Despliegue .....	86
Modelo Entidad Relación .....	87
Sub-Arquitecturas de Componentes (x Módulo) .....	88
Interacciones de Usuario .....	88
Políticas.....	92
Glosario .....	93
Bibliografía .....	94

## Control de Cambios

Fecha	Autor	Versión	Comentarios
15 de octubre de 2017	Alexander Vengoechea / Darwin Leguizamón	1.0	Reconocimiento de contenido
13 de noviembre de 2017	Alexander Vengoechea / Darwin Leguizamón	2.0	Integración y descripción de diagramas. Finalización de documento.



## Introducción

El presente documento realizará una descripción para el diseño de un sistema para despliegue de aplicaciones en ambientes Windows.

El diseño de la solución está destinado para medianas y grandes empresas que cuenten con el servicio de soporte prestado por un área de tecnologías de la información. Lo que se busca con este sistema es automatizar la instalación de aplicaciones en entornos Windows para optimizar el tiempo de respuesta frente al requerimiento de un nuevo software por parte de un usuario final, además de la necesidad de llevar el control de licencias de software que se administran en una compañía.

En la sección de arquitectura el lector tendrá una explicación más detallada de la arquitectura del sistema; se mostrarán los componentes del sistema y de conjunta interacción. Se detallarán los diagramas de componentes, despliegue, modelo entidad relación y secuencia según lo indicado por el libro [1] basados en el modelo UML (Unified Modeling Language).

## Alcance

La solución, componentes y servicios propuestos incluyen configuraciones principalmente diseñados para ambientes Microsoft Windows sobre el cual se desarrolla el proyecto. Con esto se pretende dar una solución adecuada al tema de instalaciones automatizadas bajo las condiciones propuestas en el proyecto de grado acerca de la situación actual en muchas empresas de mediano y gran tamaño.

## Supuestos

- Puede considerarse el uso de los servicios de infraestructura con otros sistemas operativos, ya que no afecta la entrega del servicio siempre y cuando el cliente mantenga su operatividad en ambientes Windows.
- El sistema estará conformado por un servidor web que se encarga de validar los permisos previamente a la descarga del instalador.
- Se cuenta con la intranet para acceso a la plataforma.
- El acceso a cada uno de los componentes se realizará a través de la red LAN de la empresa.
- El entorno de desarrollo es web.
- Se debe contar con un dominio activo para garantizar la integridad de las políticas.

## Dependencias

- Se requiere que este implementado un Controlador de Dominio con el rol de Directorio activo.
- Los equipos clientes deben tener instalado el sistema operativo Windows (a partir de la versión 7 SP3).

- El cliente ejecutara los instaladores creados por el Automatizador de Scripts.
- El almacenamiento de los instaladores estará en un File Server que contará con permisos restringidos, con acceso únicamente a la cuenta administradora del software.

#### Restricciones

- Se debe contar con un modelo de directorio activo para validación de permisos de usuario.
- Las restricciones sobre el servicio Windows Installer estará asignada a través de las GPO del directorio activo.
- El desarrollo está diseñado solo para ambientes Windows.
- El software se considera como un grupo dentro de la estructura del directorio activo con su mismo nombre, permitiendo que la jerarquía entre roles se mantenga y no se violen las GPO al momento del despliegue.
- El acceso a los instaladores se realizará por medio de protocolos SMB.
- La vista de cliente y administrador será de tipo Wizard.

#### Metodología de Pruebas

Al implementar la metodología ágil de desarrollo XP, es necesario llevar un protocolo de pruebas que se base en esta según sus fases indicadas en [2], por lo tanto se propone a realizar dos etapas las cuales son recomendadas por el libro [3] y serán detalladas a continuación.

- **Pruebas unitarias:** Es obligatorio llevar a cabo la fase de pruebas durante el proceso de desarrollo, para poder implementar cambios en el diseño si así se requiere antes de pasar a la fase de implementación. Principalmente se involucra en la fase de codificación, en donde se realizan las validaciones para cada una de las funcionalidades que fueron detalladas en los requerimientos funcionales (Véase ANEXO B – Documento de especificación de requerimientos).  
Teniendo en cuenta lo anterior, se han propuesto pruebas funcionales para garantizar cada uno de los módulos desarrollados, para ello ya se debe contar con usuarios en el ambiente empresarial simulado. Luego garantizar el rol al que pertenece cada uno y finalmente hacer las pruebas por cada elemento de la solución antes de implementarlo en el entorno de presentación.
- **Pruebas de aceptación:** Se enfoca en las pruebas de la aplicación que son llevadas a cabo en la fase de la implementación del desarrollo. Con este tipo de pruebas simplemente se mostrará al cliente interesado las funcionalidades y validaciones que realiza el sistema. Como se está desarrollando un prototipo, se validarán solamente las funciones principales que demuestran cómo el sistema cumple con las condiciones indicadas al momento de realizar el levantamiento de la información.

En resumen, la prueba final se realizará en el ambiente simulado bajo los servicios de infraestructura empresarial, para ello se debe simular una red LAN con una maquina cliente asociada al dominio, luego hacer el proceso de autenticación con usuarios de distintos roles y se procederá a realizar las mismas pruebas unitarias para garantizar el correcto funcionamiento del proyecto.

## Arquitectura del Sistema

Con respecto a la arquitectura, esta se diseña en base a los diagramas UML. Los diagramas a continuación ayudaran a entender, diseñar y especificar los elementos que conforman el sistema de despliegue automatizado de software.

### Nivel General

Para el diseño de la solución se ha decidido usar un método modular el cual nos permite resolver y acoplar cada uno de los requerimientos por fases lo que da la posibilidad de dividir cada una de las partes del proyecto en sub problemas con el método conocido como “Divide y reinaras” encontrado en el artículo [4]. Dado el concepto de la intención de usar el método, se plantea la necesidad de desarrollar este modelo por la cantidad de secciones que tiene.

El método modular exige que los módulos se interrelacionen entre sí; este principio correlaciona las distintas partes en las que se desarrolla el proyecto. Como tal el proyecto lleva a partir los requerimientos en varias partes:

Se requiere la parte lógica de la solución que se define como el despliegue de software a través de la red.

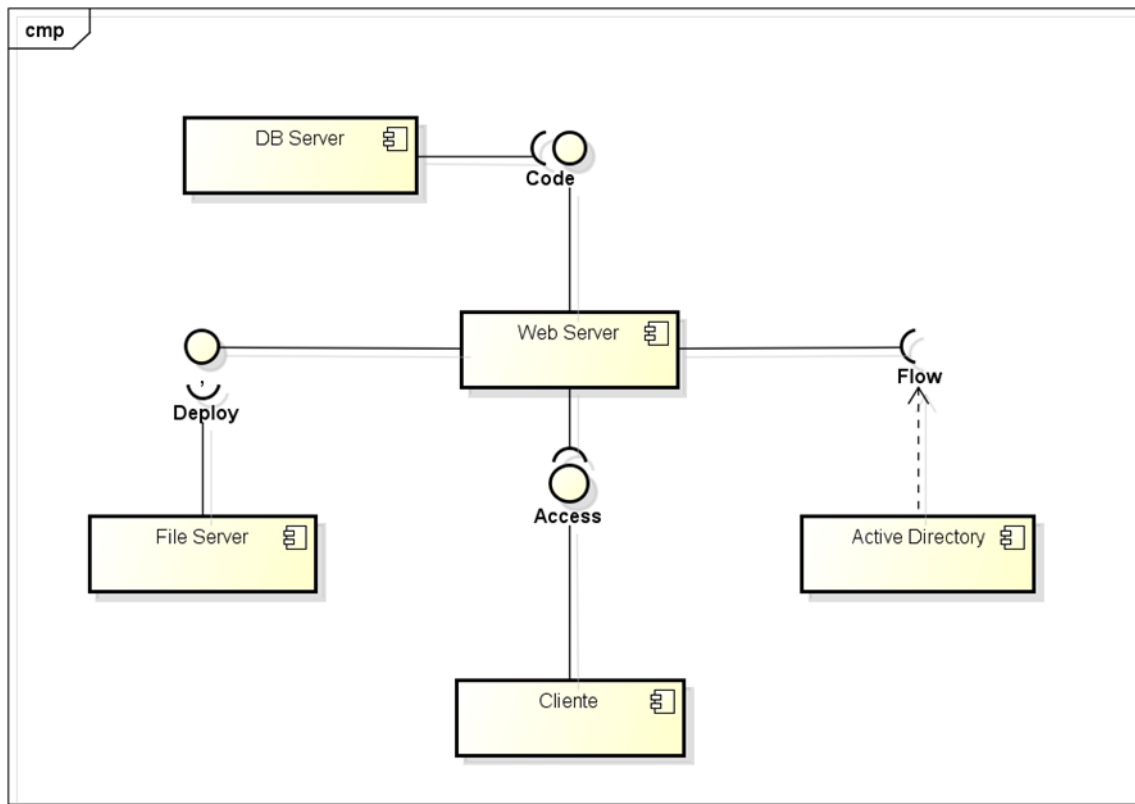
Es necesario realizar validaciones de permisos lo cual hace necesario el uso de persistencia de datos obtenidos o sincronizados con el directorio activo.

Las vistas de entorno web del sistema permitirán que el uso de la herramienta sea intuitivo. Las interfaces se realizarán de modo ayudante o Wizard para garantizar la exigencia de la facilidad de uso.

## Diagrama de Componentes

El diagrama de componentes muestra los distintos módulos que conforman el sistema propuesto, en el cual se detalla cómo interactúan los componentes en un entorno de ejecución.

**Ilustración 30 - Diagrama de Componentes**

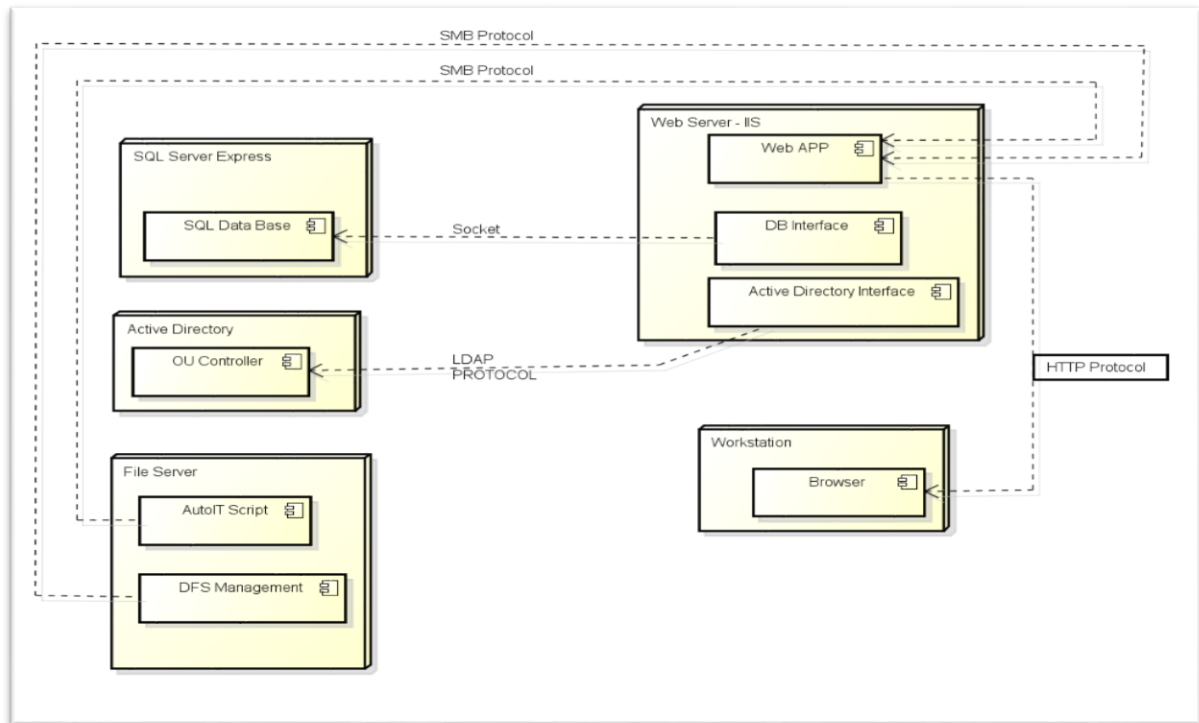


Inicialmente parte desde el componente cliente quien es el que realiza la solicitud al web server usando la interfaz de acceso, esta interfaz se encarga de validar las credenciales con el módulo de active directory en un modo directo. Ya con las credenciales validadas, el componente web server se encarga de consultar en la base de datos los permisos que el cliente tiene sobre la solicitud enviada, al ser confirmado el proceso, este se trae desde el componente File server el script de automatización con su ejecutable y de esta manera la respuesta que entrega al cliente es el instalador parametrizado.

## Diagrama de Despliegue

Con el diagrama de despliegue se pretende mostrar cada uno de los componentes de hardware y de software que componen el sistema, a su vez ilustra cómo está distribuido cada módulo en los diferentes componentes de hardware que se requieren para poder llevar a cabo el despliegue de instrucciones de automatización.

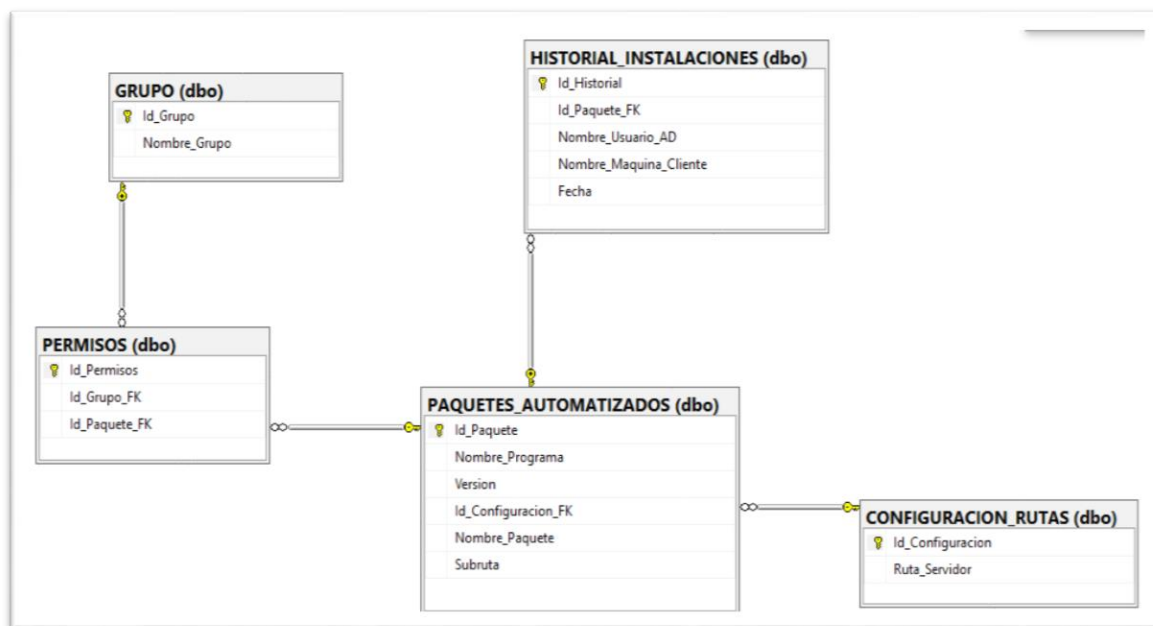
**Ilustración 31 - Diagrama de Despliegue**



El flujo comienza cuando, desde un computador (Workstation) por medio de un web browser se accede a la plataforma y se realiza una solicitud, esta por medio del protocolo http llega a la aplicación que se encuentra alojada en el servidor web. Cuando el usuario accede a la plataforma, la aplicación web valida las credenciales del usuario accediendo a una instancia del directorio activo por medio del protocolo LDAP, cuando las credenciales ya están validadas la aplicación consulta a la base de datos por medio de un socket de comunicación, los permisos que están asociados al usuario. Finalmente, cuando el usuario envía la solicitud de instalar por medio del mismo protocolo http, la aplicación se comunica con el file server por medio del protocolo SMB, en este se encuentran alojados el DFS (Distributed File System) y el AutoIT Script, entregando posteriormente la aplicación automatizada y el package (paquete) parametrizado para ejecutarse de modo desatendido como respuesta al usuario.

## Modelo Entidad Relación

Ilustración 32 - Modelo Entidad Relación



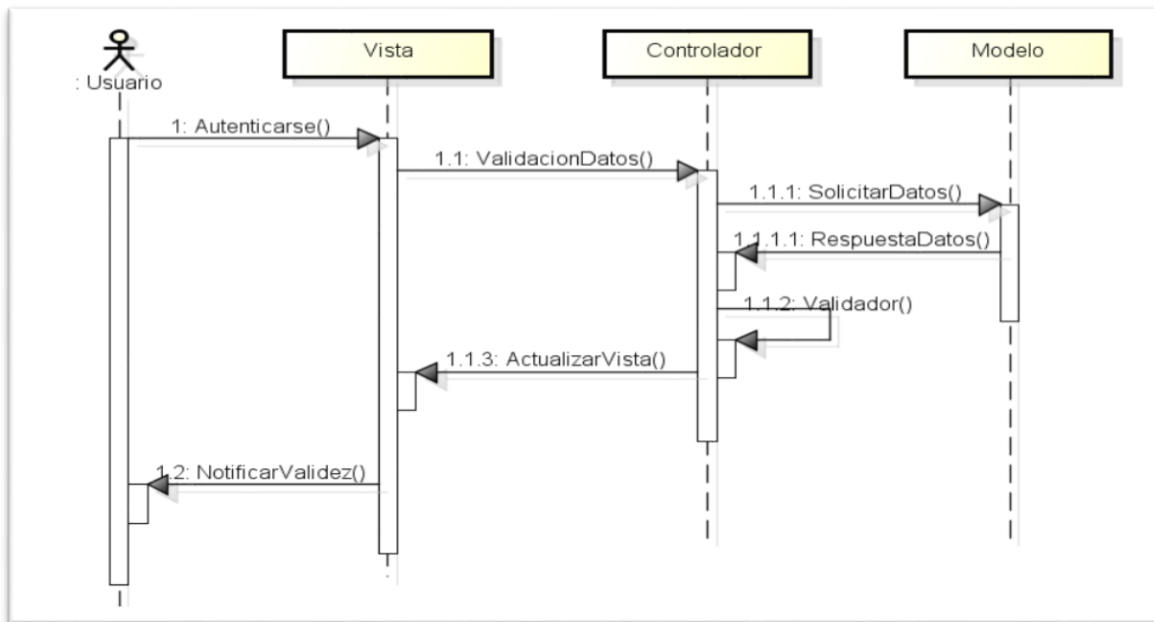
El modelo entidad relación ilustra las entidades que se consideraron importantes en el diseño de la solución y como se relacionan para garantizar el almacenamiento de los datos de una manera estable. Las entidades que se consideraron para este proyecto son:

- **GRUPO**: Tiene como finalidad almacenar el nombre del rol al que pertenecerán los programas, llamados así para mantener el control de los permisos sobre las aplicaciones.
- **PERMISOS**: cuenta con tres atributos que indican la relación que existe entre el grupo y los PAQUETES\_AUTOMATIZADOS, es allí en donde se captura la información de los permisos por paquete de instalación.
- **CONFIGURACION\_RUTAS**: cuenta con un atributo que indica la ruta en donde estarán almacenados los paquetes de instalación.
- **HISTORIAL\_INSTALACIONES**: se validó esta entidad con el fin de almacenar los registros de cada instalación.
- **PAQUETES\_AUTOMATIZADOS**: esta entidad es la tabla transaccional, allí se encuentra la información de cada instalador, asimismo debe contar con la conexión a las tablas de CONFIGURACIÓN\_RUTAS y Permisos para que las consultas tanto de permisos como de la ubicación del archivo estén siempre disponibles. Por otra parte, en HISTORIAL\_INSTALACIONES registra cada transacción que realice.

## Interacciones de Usuario

En relación con el diseño de la solución se ha propuesto el modelo MVC (modelo, vista controlador) según el artículo [5], luego es necesario enfatizar en la interacción de los distintos objetos que componen el sistema frente a su funcionalidad [1], para esto se han realizado los diagramas de secuencia que precisamente ayudaran a desarrollar este tema.

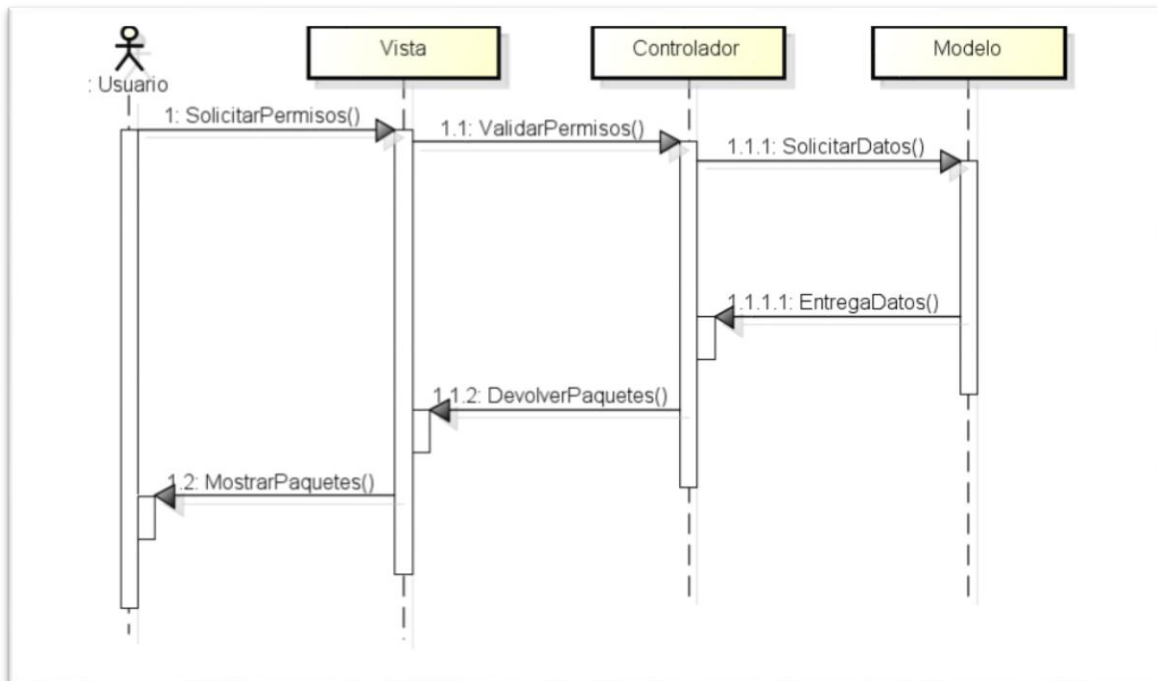
### Ilustración 33 - Diagrama de Secuencia (Autenticación)



La primera funcionalidad será la autenticación de usuarios, este modelo involucra a los usuarios normales como a los administradores. El flujo inicia con la solicitud del actor (Usuario – Administrador) a la función de autenticarse, allí en la capa de vista se ejecuta la validación realizando uso de la validación de datos a la capa controlador, la persistencia de los datos consulta con la capa de modelo a través de la función Solicitar Datos. El modelo como respuesta entrega los datos a la capa de controlador y este realiza una validación de los datos para enviar el mensaje de validación exitosa o credenciales erróneas. Finalmente cargara el perfil del usuario correspondiente.

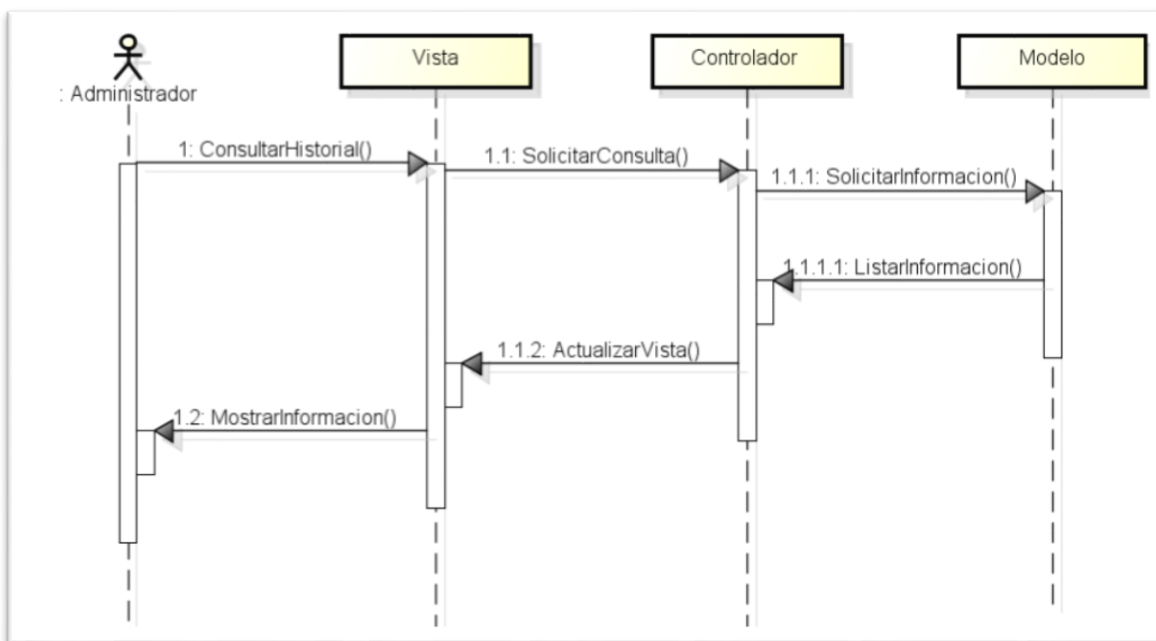


### Ilustración 34 - Diagrama de Secuencia (Despliegue)



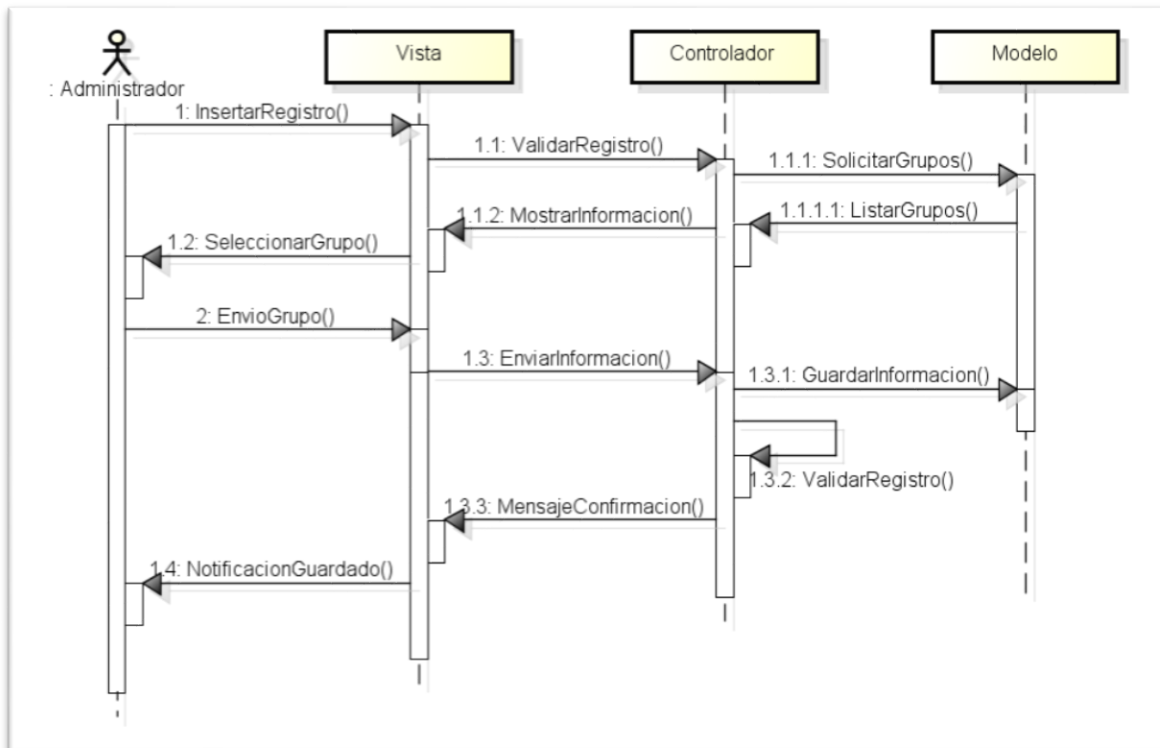
Como segunda funcionalidad se detalla el despliegue de la aplicación. Básicamente el sistema se encargará de validar los permisos a través de las capas del modelo MVC. La interacción entre estas capas inicia con la solicitud de la información del cliente al momento de proceder con una instalación, este se realiza a través de la capa de vista. Al realizar la acción simultáneamente validará los datos solicitándolos a la capa del controlador; este a su vez debe consultar en la base de datos los permisos asociados al usuario dependiendo al grupo al que pertenezca. Por respuesta el sistema enlistará de los paquetes que el usuario puede desplegar.

### Ilustración 35 - Diagrama de Secuencia (Despliegue)



La funcionalidad de consultas se hace por medio de la solicitud desde el usuario con perfil administrador, esto hace que la solicitud haga un flujo por las capas hasta consultar en la base de datos el registro de la información y por respuesta obtendrá la información detallada en un reporte.

### Ilustración 36 - Diagrama de Secuencia (Registro)



El usuario administrador es el que realiza la asignación de los permisos a los usuarios para cada software. El administrador interactúa con el sistema y a su vez este hace el proceso de validación de permisos, registros y entrega confirmación de la operación.

## Políticas

El proyecto cuenta con limitaciones académicas, lo que implica que se debe utilizar licenciamiento libre. El software libre se define en fuentes [6], como de distribución libre. También el autor explica las consideraciones que se deben tener cuando se habla de código abierto para uso libre según el usuario que lo desee implementar y modificar.

En Colombia el software libre esta cobijado por los derechos de autor, sin embargo, existe un derecho moral que implica que el desarrollo del software libre será concedido para ser adaptado, modificado y mejorado si así se requiere.

Existen principios de la propiedad intelectual que respaldan el software libre y el código propietario indicados por (artículo 187 Ley 23 de 1982) según [7], allí expresa que las obras o trabajos que sean denominados de dominio público podrán ser consideradas como renuncia a sus derechos de propiedad.

Teniendo el contexto de lo anteriormente mencionado, se procede a indicar las políticas que le competen al desarrollo del proyecto que son:

- La simulación de la infraestructura empresarial se realizará bajo licenciamiento aplicado a máquinas virtuales de uso no oficial.
- Los requerimientos y funcionalidades fueron seleccionados a partir de un proceso de consenso entre la universidad y los integrantes del grupo de trabajo por lo cual solo se entregará el prototipo de la solución.
- El prototipo y el código fuente son entregados a la universidad, acogiendo la normatividad.

## Glosario

**Ambiente:** Todo aquello que está afuera de los límites del sistema se considera como entorno o ambiente del mismo.

**Automatización:** Conversión de toda actividad repetitiva para que se desarrollen por sí solas y sin la participación directa de un individuo

**Despliegue:** Proceso por el que una aplicación informática pasa a estar lista para utilizarse.

**Directorio Activo:** Servicio de directorio en una red distribuida de computadores implementada por Microsoft

**Dominio:** Conjunto de computadoras conectadas en una red informática que confían a uno de los equipos de dicha red, la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red

**Entorno:** Sinónimo de ambiente (Véase Ambiente).

**File Server:** Tipo de servidor que almacena y distribuye diferentes tipos de archivos informáticos entre los clientes de una red de computadoras

**HTTP:** Hypertext Transfer Protocol o protocolo de transferencia de hipertextos

**LAN:** Local Area Network o red de área local

**LDAP:** Lightweight Directory Access Protocol o Protocolo ligero de acceso a directorios

**Plataforma:** Potente herramienta de gestión empresarial conformada por un conjunto de hardware

**Protocolo:** Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red

**Red:** Conjunto de dispositivos interconectados entre sí a través de un medio, que intercambian información y comparten recursos.

**Scripts:** Lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador.

**Servicios:** Conjunto de actividades que buscan responder a las necesidades de un cliente por medio de un cambio de condición en los bienes informáticos

**Servidor web:** Programa informático que procesa una aplicación del lado del servidor que provee el servicio.

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas

## Bibliografía

- [1] J. Rumbaugh, I. Jacobson, and G. Booch, "El Lenguaje Unificado de Modelado," *Elements*, p. 30, 1999.
- [2] M. C. P. P. L. José H. Canós, "Metodologías Ágiles en el Desarrollo de Software," Mar. 2012.
- [3] *Técnica administrativa*. Ciencia y Técnica Administrativa, 2002.
- [4] E. A. Posner, K. E. Spier, and A. Vermeule, "Divide and Conquer," *SSRN Electron. J.*, May 2009.
- [5] G. E. Krasner and S. T. Pope, "A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80," *Joop J. Object Oriented Program.*, vol. 1, pp. 26–49, 1988.
- [6] Z. Cataldi and F. Salgueiro, "Software libre y código abierto en educación," *Quad. Digit.*, vol. 48, pp. 1–12, 2007.
- [7] Congreso de la República de Colombia, *LEY NÚMERO 23 DE 1982 "Sobre derechos de autor"*; 1982, p. 57.

**ANEXO D – Documento de arquitectura**



**SOLUCIÓN INSTALACIONES AUTOMATIZADAS SOBRE SISTEMAS  
OPERATIVOS WINDOWS**

**DARWIN ARNULFO LEGUIZAMÓN QUINCHE      CÓDIGO: 625091**

**ALEXANDER ARTURO VENGOECHEA GUERRERO      CÓDIGO: 625084**

**DOCUMENTO DE ARQUITECTURA DE SOFTWARE**

**DIEGO ALBERTO RINCÓN YÁÑEZ MCSc**

## Contenido

Control de Cambios.....	97
Introducción.....	98
Propósito y Alcance .....	98
Definiciones, Acrónimos y Abreviaciones .....	98
Organización del Documento .....	98
Representación de Stakeholders .....	99
Definiciones de Puntos de Vista .....	99
Punto de Vista Gerencial .....	99
Punto de Vista Lógica de Negocio.....	99
Punto de Vista Técnico .....	100
Antecedentes de la Arquitectura .....	101
Antecedentes del Problema .....	101
Visión General del Sistema.....	101
Objetivos y Contexto.....	101
Vistas .....	102
Vistas Estructurales .....	102
Vista de Componentes.....	102
Capa de Utilitarios .....	103
Capa de Acceso a Datos .....	104
Capa de Negocio .....	105
Capa de Presentación .....	106
Vista de Despliegue .....	107
Glosario.....	108
Bibliografía .....	109



## Control de Cambios

Fecha	Autor	Versión	Comentarios
21 de octubre de 2017	Alexander Vengoechea / Darwin Leguizamón	1.0	Reconocimiento de contenido, redacción sin formato
14 de noviembre de 2017	Alexander Vengoechea / Darwin Leguizamón	2.0	Redacción en formato oficial, implementación de diagramas

## Introducción

Este documento pretende ayudar a modelar la arquitectura para el desarrollo de la herramienta para despliegue automatizado de software. Los puntos que se presentan a continuación se presentan bajo el modelo IEEE 1471-2000 [1].

### Propósito y Alcance

El documento SAD (Software Architecture Document) propone la descripción de la arquitectura general de la solución para el despliegue automatizado de software en ambientes Windows, paralelamente se enfocará en los aspectos de diseño necesarios para que el desarrollo de la herramienta sea de calidad.

A través de las vistas recomendadas en el formato IEEE 1471-2000, con esto se hace necesario enfocar el alcance de la solución como un prototipo en donde por medio de sus componentes logre demostrar la solides de un sistema estable que cumpla con las necesidades concertadas en el documento de especificación de requerimientos (Véase ANEXO B – Documento de especificación de requerimientos).

### Definiciones, Acrónimos y Abreviaciones

- **SAD:** Documento de Arquitectura de Software (Software Architecture Document).
- **UML:** Lenguaje de Modelado Unificado (Unified Modeling Language).
- **MVC:** Modelo Vista Controlador (Model View Controller).
- **AD:** Directorio Activo (Active Directory).
- **OU:** Unidad Organizativa (Organization Unit)
- **GP:** Política de Grupo (Group Policy)
- **LDAP:** Protocolo Ligero de Acceso a Directorios (Lightweight Directory Access Protocol).

### Organización del Documento

Este documento de arquitectura se basa en tres unidades principales las cuales se definen a continuación:

- **Introducción:** Esta unidad se centra en definir el proyecto por medio de especificaciones claras para poder dimensionar, aclarar y puntualizar las secciones posteriores.
- **Antecedentes de la arquitectura:** Presenta de un modo general el desarrollo principal de la arquitectura.
- **Vistas:** Ilustra por medio de diagramas los elementos que componen el sistema a desarrollar, enfocándolo al entendimiento de los desarrolladores y especialistas de infraestructura.

## Representación de Stakeholders

El proyecto cuenta con dos tipos que son:

- **Usuario final:** toda aquella persona que hace uso del sistema, el cual le permite realizar una instalación desatendida.
- **Usuario administrador:** toda aquella persona que preparará los registros del sistema para que el rol de usuario final pueda utilizar la plataforma correctamente. Este rol puede establecer permisos sobre paquetes de instalación y generar reportes.

## Definiciones de Puntos de Vista

Los puntos de vista se basarán en tres grandes visiones pensando en los interesados, estos son detallados a continuación:

### Punto de Vista Gerencial

El punto de vista gerencial se enfoca en el concepto de costo – beneficio, intentando dar retorno a la inversión inicial.

#### Descripción

Con el despliegue automatizado de instalaciones, el retorno de la inversión se reflejaría en tiempo por mano de obra, al tener la solución propuesta del proyecto se está liberando funciones a los usuarios de IT, que consecuentemente podrían ejercer actividades proactivas para beneficio de la empresa en menos tiempo.

#### Stakeholders

Gerente general, Gerente de tecnología.

### Punto de Vista Lógica de Negocio

El punto de vista lógica de negocio pretende dar la visión de un usuario final, en este caso sería la persona que no tiene conocimiento técnico y requiere el servicio de instalación de software.

#### Descripción

En el negocio se cuenta con la necesidad de generar el requerimiento formal antes de recibir el software que solicita, por tal motivo la eficiencia en la solicitud de este servicio sería optimizado y mejoraría la percepción del usuario frente al tipo de servicio que se está ejerciendo de forma desatendida.

#### Stakeholders

Usuarios administrativos, Gerente general, jefes de área.

## Punto de Vista Técnico

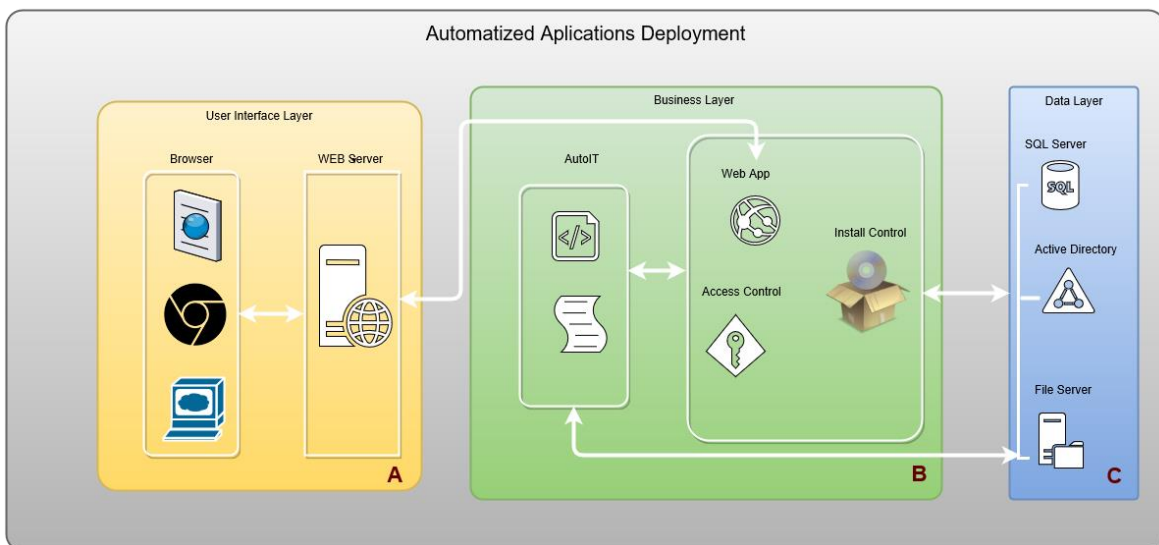
El punto de vista técnico comprende todo usuario con conocimiento técnico y tiene intereses en la solución propuesta.

### Descripción

En principio se encuentran los desarrolladores que intervienen para construir la mejor solución, de las necesidades indicadas en el levantamiento de requerimientos. Para ello se pensó en el modelo MVC que propone independencia en cada capa de sistema según el autor [2] creando así un óptimo desempeño para el entorno en el que se pretende implementar; además da la oportunidad de implementar los aspectos de diseño.

Basados en el modelo MVC, se quiere enfocar el detalle general del diagrama de arquitectura que dará más claridad al diseño de cada módulo. Teniendo en cuenta lo indicado se procede a detallar cada una de las capas MVC en el siguiente diagrama:

### Ilustración 37 - Diagrama de Arquitectura



### Stakeholders

Desarrolladores, Usuarios técnicos.

## Antecedentes de la Arquitectura

### Antecedentes del Problema

El área técnica de una compañía requiere coordinar con la persona que hace la solicitud del software a instalar, por este motivo los tiempos de respuesta se ven afectados debido a que no es por dependencia propia ese impacto. Al tener la solución activa el usuario técnico podría ejercer otras actividades gracias a la automatización de procesos que son repetitivos y mecánicos.

## Visión General del Sistema

### Objetivos y Contexto

La arquitectura de un sistema se debe basar en aspectos de diseño claros que se describirán a continuación.

- **Integración:** El diseño de la solución está pensado para dar la opción de ser integrado con otros sistemas, en este caso el sistema cuenta con una interacción directa con el AD (Active Directory).
- **Seguridad:** El sistema cuenta con un sistema de seguridad basado en autenticación por medio del AD, a su vez el entorno debe contar con roles predeterminados por grupos y de esta manera tener implementadas las GP para controlar los permisos de instalaciones. El aspecto de seguridad tiene bastante peso ya que todo el acceso a la aplicación se realizará por medio de roles puntuales sobre las OU (Organization Unit) del AD.
- **Disponibilidad:** Se contempla la necesidad de que el sistema debe estar disponible y activo cada vez que los usuarios requieren realizar procesos sobre la herramienta, sin embargo, el prototipo no tiene en cuenta aspectos de alta disponibilidad ni tolerancia a fallos.
- **Mantenimiento:** El sistema tendrá en cuenta actualizaciones con mejoras futuras que pueden ser implementadas posteriormente al desarrollo. El código es abierto y puede ser manipulado para ser adaptable a las necesidades del usuario u organización que la requiera.

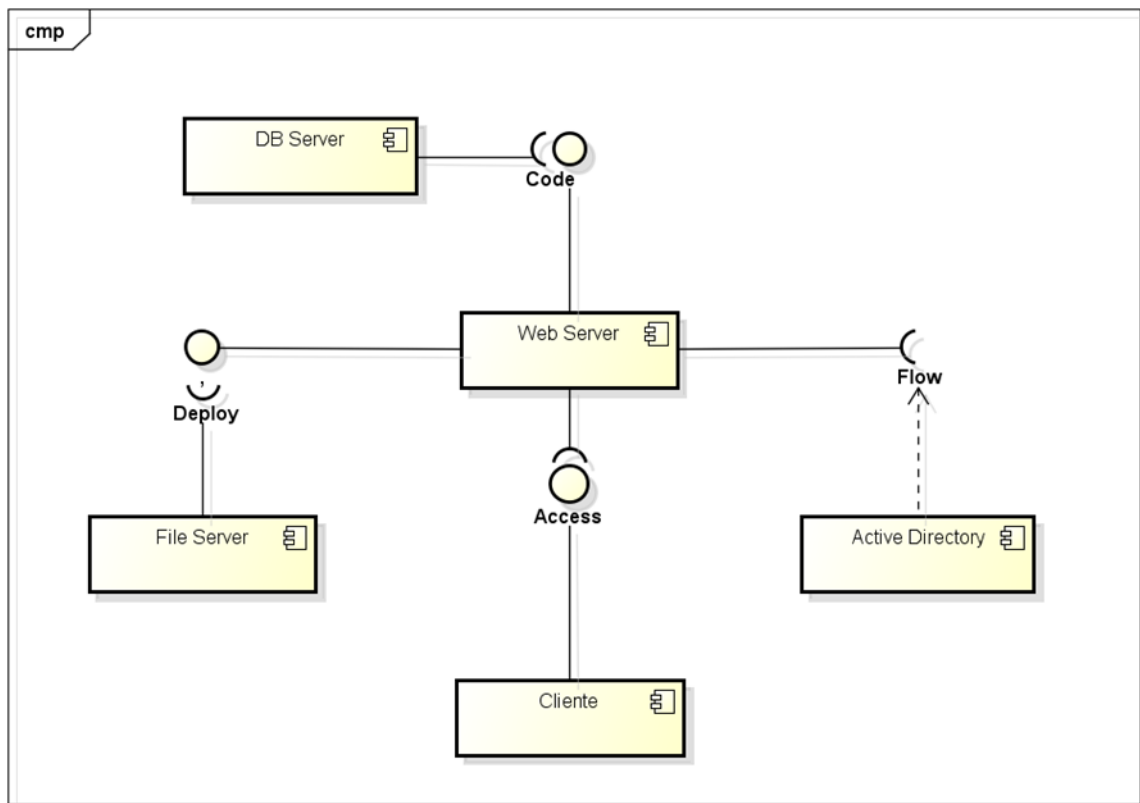
## Vistas

### Vistas Estructurales

Todo el tema de vistas estructurales es basadas en el libro [3], los detalles de cada vista se ilustran a continuación.

### Vista de Componentes

**Ilustración 38 - Diagrama de Componentes**



Inicialmente parte desde el componente cliente quien es el que realiza la solicitud al web server usando la interfaz de acceso, esta interfaz se encarga de validar las credenciales con el módulo de active directory en un modo directo. Ya con las credenciales validadas, el componente web server se encarga de consultar en la base de datos los permisos que el cliente tiene sobre la solicitud enviada, al ser confirmado el proceso, este se trae desde el componente File server el script de automatización con su ejecutable y de esta manera la respuesta que entrega al cliente es el instalador parametrizado.

## Capa de Utilitarios

La capa de utilitarios muestra cada una de las herramientas que son fundamentales para llevar a cabo el desarrollo.

### Ilustración 39 - Diagrama Utilitarios



## Capa de Acceso a Datos

En el caso de esta solución los datos son paquetes de instaladores con su correspondiente script automatizado que serán almacenados en el File Server. Al realizar el proceso se debe contar con el registro de cada despliegue, por lo cual en el modelo de base de datos se almacenará el registro de seguimiento sobre las instalaciones, así poder generar los reportes y tener el control de las licencias para cada software. La única fuente de datos necesaria en la estructura de la aplicación es el directorio activo, allí se encontrará la distribución entre grupos de forma jerárquica y sobre ellos los permisos para cada tipo de usuario.

### Ilustración 40 - Capa de Datos



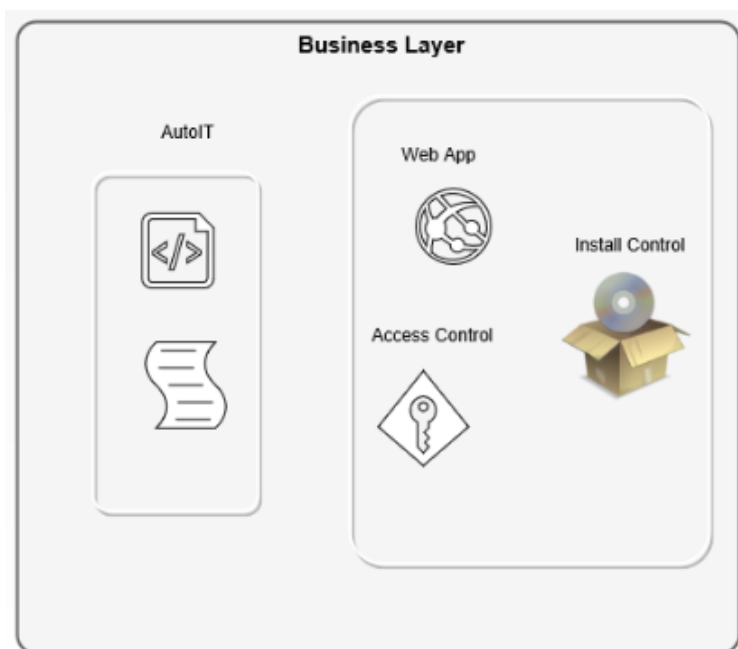


## Capa de Negocio

En la capa de negocio se encuentra toda la lógica del sistema, por un lado, está la persistencia de los datos, la cual se encarga de mantener en memoria toda la estructura de los permisos de cada usuario, este proceso es realizado desde la aplicación web. Además de realizar el proceso lógico la aplicación está encargada de mantener la integridad de los datos, como se contempló inicialmente los permisos para el uso de las aplicaciones se suministrarán a través del directorio activo, por lo tanto, la aplicación requiere realizar solicitudes continuamente al directorio activo por medio del protocolo LDAP.

Para la lógica de la aplicación, es necesario generar y almacenar scripts automatizados con las instrucciones secuenciales para cada programa que se desee instalar, es por tal razón que se implementa un módulo llamado AutoIT el cual será el encargado de esta función.

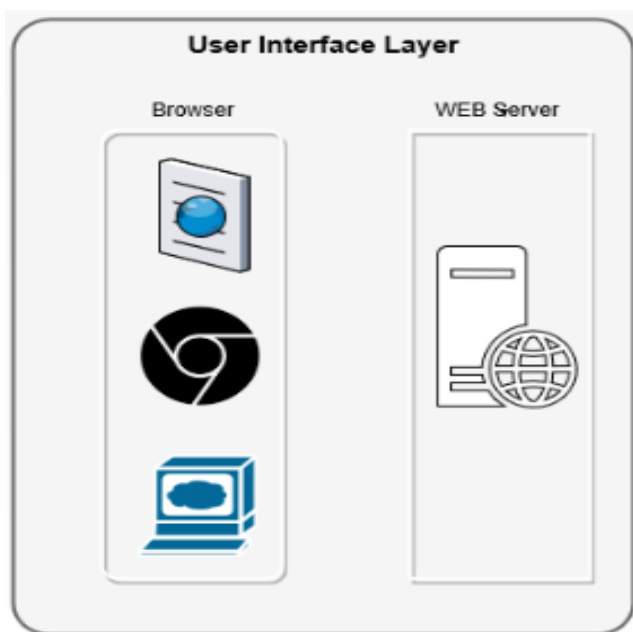
### Ilustración 41 - Capa de Negocio



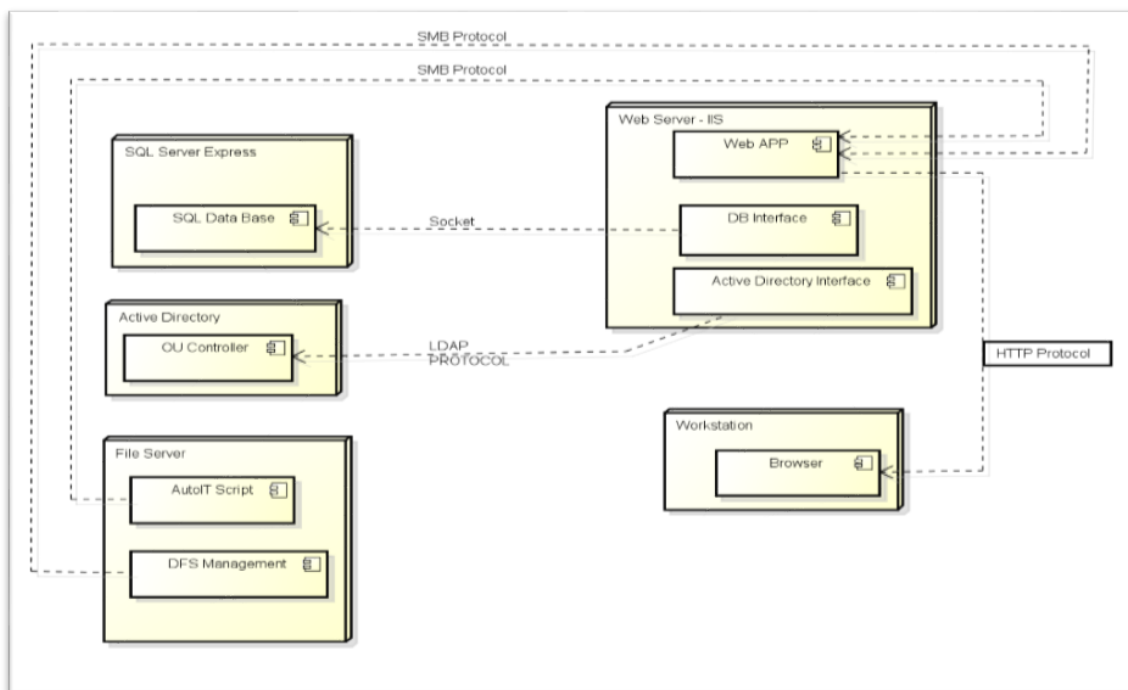
## Capa de Presentación

La capa de presentación del diagrama de arquitectura es todo lo que compete a la parte grafica de la aplicación con la cual el usuario tendrá interacción. En la parte de vista, se plantean dos tipos de elementos, el primero es un navegador que entrega la interfaz gráfica de forma amigable para los dos tipos de usuarios, los cuales se contemplaron en el documento de especificación de requerimientos (Véase Anexo B); por un lado, está el usuario normal el cual es el que solicita el software y por el otro está el administrador de la plataforma quien será el que otorgará los permisos para el despliegue de aplicaciones a los usuarios normales.

### Ilustración 42 - Capa de Presentación



### Ilustración 43 - Diagrama de Despliegue



El flujo comienza cuando, desde un computador (Workstation) por medio de un web browser se accede a la plataforma y se realiza una solicitud, está por medio del protocolo http llega a la aplicación que se encuentra alojada en el servidor web. Cuando el usuario accede a la plataforma, la aplicación web valida las credenciales del usuario accediendo a una instancia del directorio activo por medio del protocolo LDAP, cuando las credenciales ya están validadas la aplicación consulta a la base de datos por medio de un socket de comunicación, los permisos que están asociados al usuario. Finalmente, cuando el usuario envía la solicitud de instalar por medio del mismo protocolo http, la aplicación se comunica con el file server por medio del protocolo SMB, en este se encuentran alojados el DFS (Distributed File System) y el AutoIT Script, entregando posteriormente la aplicación automatizada y el package parametrizado para ejecutarse de modo desatendido como respuesta al usuario.

## Glosario

**Ambiente:** Todo aquello que está afuera de los límites del sistema se considera como entorno o ambiente del mismo.

**Automatización:** Conversión de toda actividad repetitiva para que se desarrollen por sí solas y sin la participación directa de un individuo

**Despliegue:** Proceso por el que una aplicación informática pasa a estar lista para utilizarse.

**Directorio Activo:** Servicio de directorio en una red distribuida de computadores implementada por Microsoft

**Dominio:** Conjunto de computadoras conectadas en una red informática que confían a uno de los equipos de dicha red, la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red

**Entorno:** Sinónimo de ambiente (Véase Ambiente).

**File Server:** Tipo de servidor que almacena y distribuye diferentes tipos de archivos informáticos entre los clientes de una red de computadoras

**HTTP:** Hypertext Transfer Protocol o protocolo de transferencia de hipertextos

**LAN:** Local Area Network o red de área local

**LDAP:** Lightweight Directory Access Protocol o Protocolo ligero de acceso a directorios

**Plataforma:** Potente herramienta de gestión empresarial conformada por un conjunto de hardware

**Protocolo:** Conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red

**Red:** Conjunto de dispositivos interconectados entre sí a través de un medio, que intercambian información y comparten recursos.

**Scripts:** Lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador.

**Servicios:** Conjunto de actividades que buscan responder a las necesidades de un cliente por medio de un cambio de condición en los bienes informáticos

**Servidor web:** Programa informático que procesa una aplicación del lado del servidor que provee el servicio.

**Software:** Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas

## Bibliografía

- [1] IEEE-SA Standards Board, "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," *IEEE Std*, vol. 1471–2000, pp. 1–23, 2000.
- [2] G. E. Krasner and S. T. Pope, "A Cookbook for Using the Model- View-Controller User Interface Paradigm in Smalltalk-80," *Joop J. Object Oriented Program.*, vol. 1, pp. 26–49, 1988.
- [3] C. B. Reynoso, "Introducción a la Arquitectura de Software."